Automatic Reconstruction From Serial Sections

Elizabeth Guest

Ph.D.
University of Edinburgh
1994

Abstract

In many experiments in biological and medical research, serial sectioning of biological material is the only way to reveal the three dimensional (3D) structure and function. For a number of reasons other 3D imaging techniques, such as CT, MRI, and confocal microscopy, are not always adequate because they cannot provide the necessary resolution or contrast, or because the specimen is too large, or because the staining techniques require sectioning. Therefore for the foreseeable future reconstruction from serial sections will remain the only method for 3D investigations in many biomedical fields. Reconstruction is a difficult problem due to the loss of 3D alignment as the sections are cut and, more seriously, the systematic and random distortion caused by the sectioning and preparation processes.

Many authors have reported how serial sections can be registered by means of fiducial markers or otherwise, but there have been only a few studies of automated correction of the sectioning distortions. In this thesis solutions to the registration problem are reviewed and discussed, and a solution to the warping problem, based on image processing techniques and the finite element method (FEM), is presented. The aim of this project was to develop a fully automatic method of reconstruction in order to provide a 3D atlas of mouse development as part of a gene expression database. For this purpose it is not necessary to warp the object so that it is identical to the original object, but to correct local distortions in the sections in order to produce a smooth *representative* mouse embryo. Furthermore the use of fiducial markers was not possible because the reconstructions were from already sectioned material.

In this thesis we demonstrate a new method for warping serial sections. The sections are warped by applying forces to each section, where each section is modelled as a thin elastic plate. The deformation forces are determined from correspondences between sections which are calculated by combining match strengths and positional information. The equilibrium state which represents the reconstructed 3D image is calculated using the finite element method. Results of the application of these methods to paraffin wax and resin embedded sections of the mouse embryo are presented.

Acknowledgements

This thesis would not have been possible without the help and support of my three supervisors: Richard Baldock (MRC Human Genetics Unit) for support throughout my period of study and for providing a few hundred megabytes of disc space for my results, Jim Piper (MRC Human Genetics Unit) for much help in the early stages, and John Hallam (Department of Artificial Intelligence) for liaison with the University and general support when needed. I would also like to thank Glynn Robinson (Department of Diagnostic Radiology, Yale University) and Lewis Griffin (Department of Neurology, UMDS, University of London) for providing the code to produce a Delaunay triangulation from a list of points, and Daniel Cunliffe, my office mate, for always being prepared to listen to ideas and check extracts of documents for comprehensibility.

The resources for this project were provided by the MRC Human Genetics Unit (Western General Hospital, Crewe Rd, Edinburgh), and funding was provided by the SERC, the University of Edinburgh, and my parents (with the help of the Dutch Government).

Finally I would like to acknowledge the members of Struthers Memorial Church, Edinburgh for their love, support and encouragement throughout my studies.

Declaration

I hereby declare that I composed this thesis entirely myself and that it describes my own research.

E. A. Guest
Edinburgh
December 19, 1994

Contents

List of Figures

Chapter 1

Introduction

Techniques for solving problems associated with image analysis and interpretation are important in biomedical research due to the increasing use of digital image capture facilities. In this thesis image processing techniques have been developed to solve the problem of automatic three dimensional reconstruction from serial microtome sections.

Slicing a specimen into thin sections is a routine technique which has been used since the end of the last century to investigate the three dimensional structure of biological objects. The specimen is embedded in rigid material, such as wax or plastic, and sectioned using a microtome. The sections are then floated on water to flatten them out before being placed on a microscope slide and stained. The process of mechanical sectioning followed by handling of the sections arbitrarily shifts and rotates each section and also introduces a number of distortions. To reconstruct the original object we need to *register* and *warp* each section. Registration involves correcting the rotation and translation errors; warping involves correcting the distortions.

The first reconstructions were done manually by transferring the outlines of the structures onto wax or plastic plates and then stacking these plates. Since the mid 1960's there have been attempts at automating the process, but these have mostly been limited to producing interactive methods for registering the sections and to producing sophisticated displays. More recently work has been done on automating the registration, but very little work has been aimed at correcting the distortions.

In many cases, other techniques such as CT scans, MRI microscopy, and confocal laser scanning microscopy can be used to obtain 3D images of structures directly without

the need for reconstruction. However for many applications requiring resolutions of a few microns, serial microtome sections remain the only option. For these resolutions confocal microscopy is the only alternative but this requires fluorescent or reflecting specimens and, because of limited light penetration depth, it is not suitable for specimens more than about 100 microns thick. Staining requirements may also be a factor as stains which are suitable for confocal microscopy, in that they provide the necessary fluorescent properties and can be applied evenly to the whole specimen, may not show up all the required structures.

Automatic reconstruction of serial sections is a difficult problem because 3D adjacency information is lost when the sections are cut, and because of distortion. The loss of the 3D adjacency information may mean that the original object cannot be reconstructed. For example in this way a helix can be reconstructed as a cylinder, and if there is no limit on the amount of warping allowed, a sphere could also be reconstructed as a cylinder. To reconstruct the original object additional information is required such as fiducials or images of the block face before each section is cut. In this thesis we restrict ourselves to the problem where serial sections without fiducials are given. The specific application domain is that of reconstructing a 3D atlas of mouse development as part of a gene expression database which will provide a resource for analysing gene expression in developing mouse embryos [6, 83]. For this purpose it is not necessary to warp the object so that it is identical to the original object but to correct local distortions in the sections in order to produce a smooth *representative* mouse embryo. The use of fiducial markers was not possible because most of the required reconstructions were from already sectioned material used for an existing "paper" atlas [55].

A major goal of the work described in this thesis is to automate the reconstruction process. A fully automatic reconstruction is required because of the operator time involved in interactively registering and warping each of thousands of sections (the atlas of mouse development may involve 20 thousand sections). Many authors have published methods for registering sections, but few have attempted to solve the warping problem. The focus of this work was to develop methods for warping serial sections and was achieved by modelling the sections as thin elastic plates which are warped by the application of forces calculated by matching to adjacent sections.

An illustration of the problem and the effects of the warping method developed here is shown in figure 1.1 where a 3D image of a 9 day mouse embryo before and after warping is shown. The top surface of each block shows a section taken in the original cutting direction; the other surfaces show orthogonal sections and a section cut in an arbitrary direction. From these images we see that warping has significantly improved the quality of the arbitrary section through the stack, and indeed, after warping the quality of an arbitrary section is similar to that of a section taken in the original cutting direction, showing that the embryo has been satisfactorily reconstructed.

In order to correct local distortions, correspondences between adjacent sections must be found. Clearly, the search space for finding correspondences can be dramatically reduced if the sections are (roughly) registered because this enables the search to be restricted to local neighbourhoods. In addition, warping is computationally more expensive than registration and therefore to reduce computational cost, it would be helpful if the sections could be registered prior to warping. Several automatic registration algorithms have been presented in the literature and some of these were implemented. When their performance was evaluated with respect to our sections, it was found that some were suitable for our purposes and therefore the deformation correction process can assume that the sections are registered.

Deformation correction was achieved by using the finite element method to calculate a numerical solution to the equilibrium equations obtained by applying forces to thin elastic plates which are each constrained to deform only in the plane of the plate. This is the first time the finite element method has been applied to the problem of warping serial sections. There are two parts to the force calculation, which forms a major part of the thesis: matching features between adjacent sections, and using the resulting matchmaps to calculate the forces to apply to the sections. A number of existing methods of feature extraction and matching were tried, but were found to be inadequate for this application domain. Therefore a new image filter which calculates the mean and standard deviation on either side of an hypothesised line was developed, and a new matching algorithm was devised based on these feature attributes. Calculating forces is difficult due to the conflicting requirements of correspondence and warping. Since we may assume that the sections are registered and that therefore the search for correspondences can be

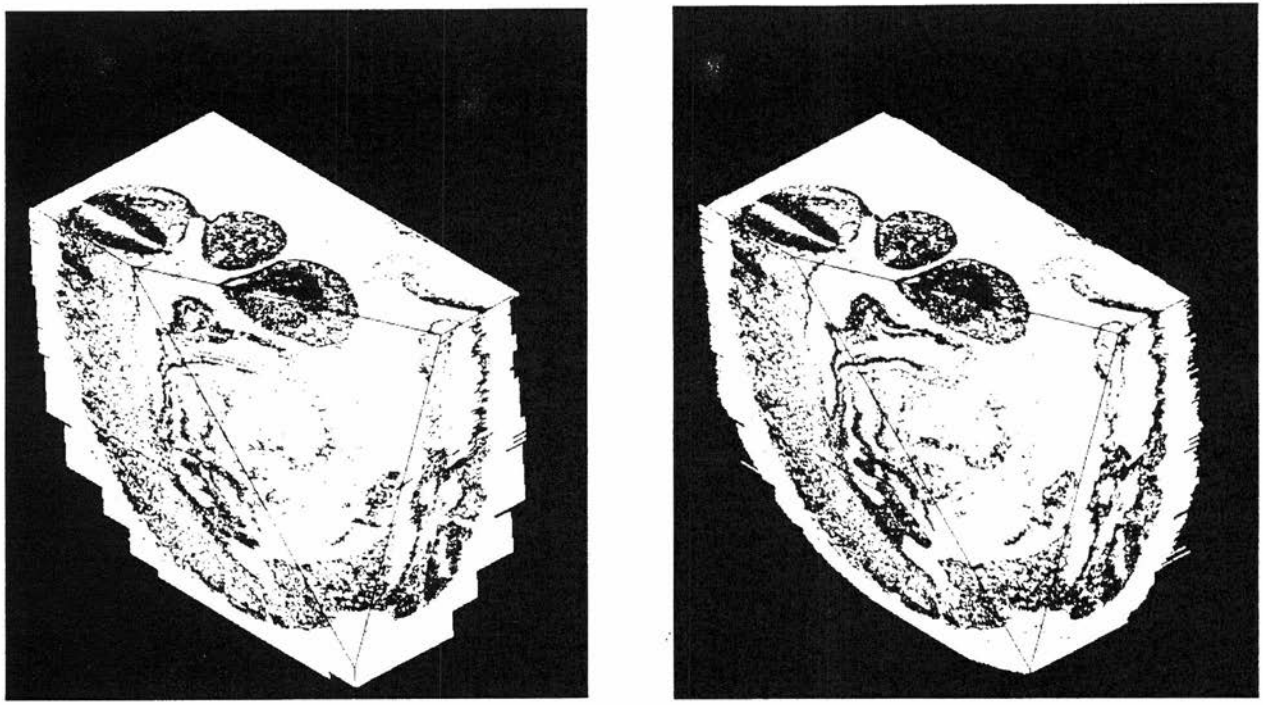


Figure 1.1: Images of a (manually registered) 9 day mouse embryo before (left image) and after (right image) warping. The top surface of each block shows a section taken in the original cutting direction; the other surfaces show orthogonal sections and a section cut in an arbitrary direction.

restricted to local neighbourhoods, we may require that more weight be given to closer and better matches as these are more likely to be correct. For warping we assume that given three corresponding points, one on each of three adjacent sections, the desired position of the point on the middle section is midway between the positions of the other two points. To achieve this, forces applied by more distant correspondences must be stronger than those applied by closer correspondences. This behaviour can be achieved by modelling the forces as springs where the spring constants depend on the likelihood that the correspondences are correct. However the likelihood that a correspondence is correct must not depend on the distance between two corresponding points as this would cancel out the effects of longer springs. To allow the distance between candidate corresponding points on adjacent sections to influence the choice of correspondence, but not contribute directly to the weights assigned to the springs, the weights assigned to the springs were made to depend on the sensitivity of candidate matches to movement of the point being matched. Although the finite element method can be used to warp individual sections, it is computationally infeasible to warp all the sections simultaneously. Therefore an iteration scheme was devised for warping the whole stack. This involves warping alternate sections on each pass through the stack, where one iteration involves two passes. The process stops when the change after an iteration is sufficiently small. Finally, a measure of success for comparing the results of warping the same stack of images using different parameters had to be developed because correlation, the measure generally used in the literature, is inappropriate. This new measure is based on a measure of the 3D structure represented by the serial section images.

Although a specific image analysis problem is addressed in this thesis, it is envisaged that the techniques developed will be applicable to other image analysis applications. In particular, the methods will be directly applicable to the problem of warping 2D images onto an atlas, and can be extended to full 3D warping. This would introduce new and interesting problems.

1.1 Overview

An introduction and review of the background information for this thesis can be found in chapters 2 and 3. In chapter 2 we give a brief history of reconstruction methods before discussing the methods of preparing and sectioning the specimens, and the problems posed by reconstruction. After the literature review in chapter 3, we discuss methods of registering the sections. Some of the earlier algorithms from the literature have been implemented and their performance compared.

The remainder of the thesis is devoted to warping. We first discuss the approach to solving the warping problem and then introduce the problems associated with matching, correspondence finding, and the force calculation. The solutions to these problems are discussed in chapters 7 and 8. Chapter 6 is devoted to the finite element method and outlines the techniques for both linear and non-linear finite element analysis.

In chapter 9 we discuss the problems of evaluating the results of the experiments described in chapter 10 and we present and evaluate a new measure of the quality of a reconstruction. The warping algorithm is tested and evaluated in the context of mouse embryology, and the results are presented in chapter 10. Finally aspects of the algorithm which can be improved and future directions of this work are discussed in chapter 11.

1.2 Definition of Terms

Terms which describe processes, like 'registration', and properties of parts of the sections, such as 'deformation' and 'feature' are used differently by different authors. In the literature, registration and alignment are used to describe both affine and non-linear transformations, depending on the context. In this document, to avoid confusion, *registration* will only refer to rotation and translation; *alignment* will be reserved for bringing individual features into line with each other, and may involve non rigid transformations. Some authors use deformation to refer to image transformations, but it is also often used to describe the distortion in the sections. Here deformation will only be used to describe the distortion of the sections; deformation correction will be referred to as *warping*.

The term 'feature' is often used to refer to specific structures which can be segmented from the images such as lines, edges, boundaries and regions. It is also used to refer to measurements which can be derived or measured from the image or image structures such as centroids and principal axes; or lengths, areas, and compactness. Three dimensional structures are also referred to as features in a three dimensional image. Therefore the term feature is used to describe a variety of different structures and properties of the image. To avoid confusion about what is meant by a feature, we use the following definitions. The term *feature* will refer to any image structure which is useful in registering and warping the sections. Features will, in practice, consist of contours, lines, points or regions which have been extracted from the image in some way. *Contours* are the boundaries of (anatomical) structures. Any measurements such as, for example, the mean and standard deviation of the pixels in a region, which can be taken from the features defined above, will be referred to as *feature attributes.* The term *object* will refer to a structure in an image which is completely surrounded by background, or to a collection of such structures. All the tissue contained within a single section will also be referred to as an object.

A summary of the definitions of these terms is found in the glossary.

Chapter 2

Discussion of the Reconstruction Problem

Three dimensional reconstruction of serial sections is an important tool in research aimed at understanding the 3D structure and function of biological material. In this chapter we give a brief history of reconstruction methods, and discuss why alternative methods for studying the 3D structure of biological objects are not suitable for this application. The mechanical details of sectioning as well as a brief review of the results of studies aimed at estimating the tissue distortions caused by the sectioning process are then discussed. Finally the problems which a reconstruction method must overcome for the purposes of a representative anatomical atlas are discussed.

2.1 Background

Three dimensional reconstruction of serial sections dates back to the second half of the 19th century (see Gaunt and Gaunt [31] for a detailed historical review). The aim is to find out what various biological structures look like internally as a prerequisite to understanding their function. To reveal the full internal 3D structure, the specimens were cut into slices/sections and then reconstructed. At first the reconstructions were modelled freely in clay using the slices as a guide. Then in 1883 Born pioneered the use of the wax plate technique where the outlines of the structures in the sections are transferred onto wax plates and then cut out and stacked. The surface of the resulting model could then be smoothed. Since then several techniques have been developed and

refined. Models can be made out of stacks of wood, or plastic, for example, as well as wax. If transparent plastic sheets are used, the outlines of the structures can be drawn onto the plastic sheets and when the plastic sheets have been assembled, it is possible to see the full internal 3D structure of the object. Several projection drawing methods have also been developed. These can give detailed views of any part of the object from several different angles (see Gaunt and Gaunt 1978 [31] for more details).

X ray CT and magnetic resonance imaging (MRI) can be used to obtain sectional views of an object without actually cutting it. X-ray CT uses X-rays to obtain 1D projections of a 2D section of an object for a series of angles from 0 to 180 degrees. These projections can be transformed (Radon transform) to form a clear 2D image of the section [64] and these are then stacked to form a 3D voxel image. MRI involves placing the subject in a strong magnetic field to align the nuclear spins. A radio frequency signal is applied which displaces some of the nuclear spins from their aligned position. When the radio transmitter is switched off, the nuclear spins emit a radio frequency signal as they return to their aligned positions. These signals are detected and used to form a three dimensional image of the subject. The maximum resolution for CT scans is 1mm [57], so it is not suitable for studying microscopic structures; in contrast, MRI microscopy is now becoming an accepted method for studying microscopic structures in applications which require a resolution of a maximum of 10 microns [50, 53], though the problem of staining to enhance the tissues of interest remains.

Another method for studying microscopic objects is confocal microscopy. A confocal microscope uses a point source of light and a point detector behind a pinhole to scan the specimen. The point source of light is focused to a small spot of light in the focal plane of the objective lens and the same lens is used to focus the point detector onto this spot of light. This makes out-of-focus regions dimmer than the focused region. Each plane in the specimen is scanned by the spot of light and the detector synchronously to build up a three dimensional image. This method is not suitable for all biological specimens because it is essential that the tissue is transparent and has good fluorescence or reflectance properties. In addition because of limited light penetration, it is not suitable for specimens more than 100 microns thick.

These methods of virtually sectioning objects have the advantage that correspondences

between features in adjacent sections are maintained, simplifying the reconstruction problem. But virtual sectioning techniques are unsuitable for some studies. This is because the particular application may require higher resolution than is possible with CT or MRI scans, but lower resolution than that provided by the electron microscope (high voltage electron microscopy allows virtual sectioning). In this case confocal microscopy is the only alternative to sectioning, but this may not be suitable because the specimen is too dense or too thick to see through, or because the required stain for the particular application does not provide the necessary fluorescence or reflectance properties. In some cases the required stain needs good access to all parts of the specimen which can only be achieved by sectioning. Because of these reasons, techniques for reconstructing (cut) serial sections remain important for the foreseeable future and are an active research field.

The particular application considered in this work is the problem of reconstructing mouse embryos for a database of mouse development [6, 83]. This requires serial sectioning because resolutions of a few microns are required and the specimens are unsuitable for confocal microscopy due to their size and staining requirements. In addition, it is desirable to use the sections from the existing Atlas of Mouse Development produced by Kaufman [55]. For this application the requirement is a representative mouse embryo at each stage which does not have to be a faithful reconstruction of the originally sectioned embryo.

The main problem with reconstructing cut serial sections as opposed to virtual sections is that the sections are displaced and deformed by the preparation process. Because of this, they have to be *registered* to correct the displacements and *warped* to correct the deformations. The registration problem can be solved manually using a 'best fit' method: Two sections are superimposed in some way and one of the sections is rotated and translated until the correspondence between the sections is judged to be maximal. The superposition can be achieved by, for example, tracing the features onto clear, plastic sheets, or by projecting one image on top of the other. A more objective method is the optical shuttle method. For this, a piece of equipment is needed which can both superimpose the two images and alternatively illuminate each image (one image is shown to each eye). One of the sections is rotated and translated to reduce the

flicker in the image which occurs when the images of the two sections are illuminated alternatively. Carlbom *et al.* [19] have implemented a version of this method which they call an "iterative digital blink comparator". The warping problem is a lot harder and, other than methods based on smoothing reconstructed surfaces, there have been few attempts at solving it. In fact without automation, warping other than by selected surface smoothing would not be cost effective. The number of sections for a single specimen may be up to 1500, and to mark sufficient tie points to correct warping would require many man months of tedious work. For the mouse development atlas there will be an estimated 20,000 sections which will need registration and warping.

Computers have been used as an aid for three dimensional reconstruction of serial sections since 1965. At first the early packages concentrated on 3D display and on surface generation. It was typically left to the user to register the sections and to pick out and label the features of interest. The computer could then stack the contours and generate a 3D impression of the reconstructed surface. The display packages have become fairly sophisticated with the use of hidden line removal, shading to give depth cues, various editing facilities (for example cutting away parts of some surfaces to reveal the surfaces underneath), and rotating the reconstructed surface.

More recently, some researchers have been attempting to automate the registration and warping of the sections and some progress has also been made towards automating feature detection. In this document the emphasis will be on the registration and warping of the sections and a review of published registration and warping algorithms is included in chapter 3.

Biologists have found computer packages very useful as an aid to their research [38] [62], although some morphological detail can be lost in computer reconstructions. Computer packages speed up the reconstruction process and produce reasonable three dimensional impressions of objects. This also makes three dimensional reconstruction more accessible as there is no need for drawing and modelling skills. Furthermore, the computer display is more flexible than hand-made models or drawings: not much can be done with a stacked series of plates, but arbitrary sectioning and editing is possible with a digitally generated display. With these displays, it is much easier to go back to the original data either to check the reconstruction or to repeat it. An important advant-

age of reconstruction by computer is that the specimen can be sectioned in a different direction to the original cutting direction. This advantage is not realised by most reconstruction packages as they do not reconstruct the original image: they just reconstruct the surfaces of the different objects.

It is important to automate feature detection, registration (and warping when considered) of the sections as it is very tedious to recognise and label structures by hand. Manual or interactive registration and warping is prone to error especially if a "best fit" method is used because the operator may be influenced by more prominent features causing these to be aligned at the expensé of other features in the images. Ideally, fiducial markers should be included in the sections to aid in the registration process but this is not always possible. In principle fully automatic reconstructions will be more objective and accurate. An important advantage of computer reconstruction is that it is a lot easier to use the computer to make morphometric measurements such as, for example, lengths, surface areas, cross section areas and volumes from the reconstruction as well as stoichiometric measurements such as optical density.

2.1.1 Preparation of Specimens

The mechanical process of preparing and sectioning the specimens causes displacements and deformations in the sections. See Gaunt and Gaunt [31], for more information.

The preparation process is similar whether the specimen is a block of tissue as small as a cubic millimeter, or as big as a whole brain. The first thing to do is to fix the specimen: treat it to stiffen the structures so that they do not collapse and to preserve the specimen. If the specimen is small it is better to remove it from the surrounding tissue after it has been fixed to minimise the deformation caused by removing it. This can be further reduced by including some of the surrounding tissue. An additional problem is that the fixing process tends to make the specimen shrink because of dehydration. This can make it slightly easier to identify the different objects within the specimen, because it tends to separate the surfaces, but it can also cause some tissue to break away from other tissue [9]. The problem is that not all tissues undergo the same amount of shrinkage and the difference may be as great as 30% [84].

After fixing, the specimens have to be embedded in more rigid material such as, for example, wax, celloidin, araldite, paraffin, or they can be frozen in gelatine [97, 98]. Fiducial markers may be embedded with the specimen to aid the registration process. A number of different objects have been used as fiducials: Toga et al [97, 98] freeze tubes of brain matter with the whole brain; other researchers use nerve fibres, blades of grass, slices of liver or brain, or the specimen can be stuffed up a chive leaf of suitable diameter. The specimen is held in place by fine hairs inside the leaf and the structures of the leaf can be used to register the sections. There are other methods of incorporating fiducials into the sections: drill a pattern of holes (with a laser [58]) through the block or paint two sides of the block [66].

A microtome is used to slice the block. As the sections are usually thin (a few microns), the microtome is very sensitive to any movement or vibration. Even air movement caused by the movements of the operator can cause variations in the thickness of the sections. It is also important to cut all the sections without stopping the knife in order to minimise the loss of tissue. The microtome produces ribbons of sections which are floated on water to remove wrinkles. The ribbons are transferred to the microscope slide via a drop of water. At this stage some of the sections may be lost or they may get out of order. In fact it is very difficult to produce a perfect series of sections with no tears or folds, and several specimens may have to be sectioned before a satisfactory series is obtained. When the water has evaporated the sections are stained to bring out the structures of interest. It is hard to stain all the sections uniformly and this can lead to differences in the greylevels of structures after digitisation.

It is important to treat all the sections in the same way to minimise variations in thickness, staining and deformation. The amount and type of deformation undergone by the sections depends on a number of factors including the type of tissue, the embedding medium and how well it penetrates into the tissue, the thickness of the sections, the temperature of the water during the "floating out" stage, the sectioning direction relative to the tissue, and the type of knife used. In recent studies [10, 23, 51], it has been found that plastic sections tend to experience compression in the cutting direction and expansion perpendicular to the cutting directions. Wax sections tend to experience expansion in all directions, but more expansion occurs perpendicular to the cutting

direction. This is probably because compression caused by cutting the sections is not fully corrected during the floating out stage. The overall expansion is due to partial melting of the wax. It was also found that the sections within each block can show variations in their dimensional changes implying that any distortion correction scheme must be able to correct each section on an individual basis [51]. The actual dimensional changes observed varied from 0-15% for plastic sections containing nerve fibres [10] and 0-25% for wax sections containing muscle fibres [51]. In some studies progressive trends due to blunting of the knife were observed [104, 51].

The viewing process can cause further deformation. If the electron microscope is used, for example, some tissue may be deformed as a result of heating under the electron beam. Further deformation can arise as a result of the digitisation process, and there may be inaccuracies in the mechanical/optical aspects of obtaining the images. For example, the image may be deformed by the microscope lens, the illumination may not be uniform and the digitisation will inevitably introduce noise.

2.2 Problems Affecting the Reconstruction

When an object is cut into sections, some of the three dimensional information is lost. Most, but not all, of this information can be recovered. Even when the sections are not deformed, the reconstruction problem is not trivial because the sections can give ambiguous information. When deformations are present, there are the additional problems of finding the deformations and working out how to warp the sections to recover the original shape.

First consider the case where the sections are not deformed. In this case, only rotations and translations will be needed to reconstruct the original image. When an object is sectioned, the correspondences between features in different sections are lost. A lot of these correspondences are recoverable, but in some situations, the information cannot be recovered and there will be alternative ways of reconstructing the object. It is possible to section two different objects and obtain identical sections. Examples of such objects include a prism and a prism with a twist; or a cylinder and a spring (figure 2.1). The prism is an example of the case where the relative rotation of the sections cannot be

Figure 2.1: A spring can also be reconstructed as a cylinder.

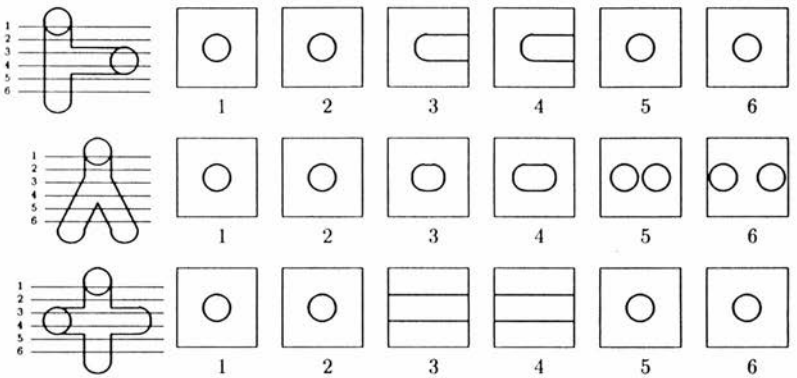


Figure 2.2: Sections through branching structures.

recovered. In the case of the spring, the relative translation is not recoverable. In both of these cases, there are an infinite number of ways of reconstructing the objects.

Generally, features will not change much between sections, but sudden changes in size and shape will occur. Features will also appear and disappear as the boundaries of objects are reached. Branching structures are another source of sudden changes between features in adjacent sections. In such cases, a single feature will have two or more corresponding features in an adjacent section; or the corresponding feature will have a very different size and shape which will continue for a few sections before returning to a size and shape similar to the original feature. See figure 2.2 for examples of branching structures and Haig *et al.* [37] for a detailed discussion of the problems of branching structures in the context of matching contours.

It can be assumed that a specimen will be sectioned perpendicular to its long axis. But a specimen will contain several structures, not all of which will be conveniently placed so that one of their major axes is perpendicular to the cutting direction. In this case corresponding features will seem to move between sections. For an example of

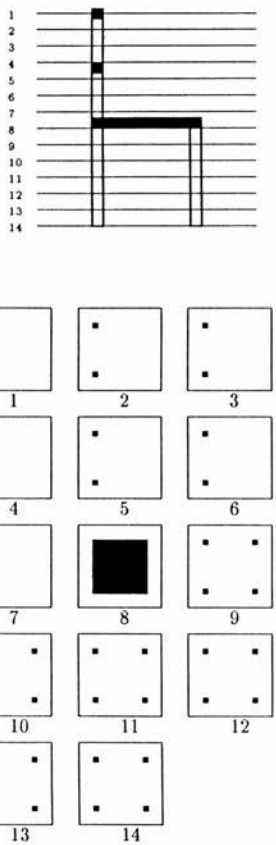


Figure 2.3: A chair sectioned horizontally.

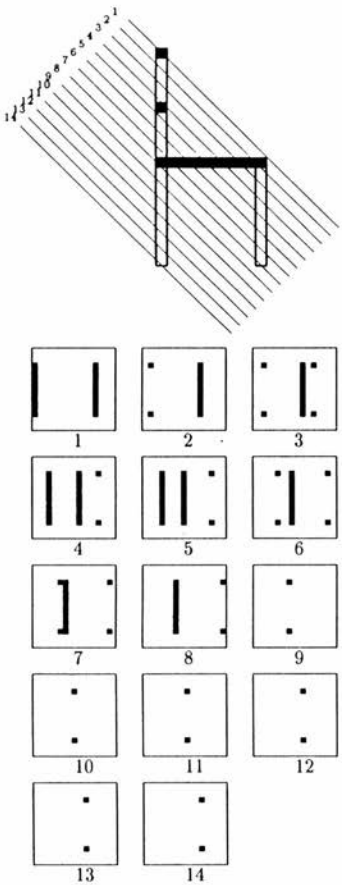


Figure 2.4: A chair sectioned diagonally.

this, compare figures 2.3 and 2.4, where a chair is cut horizontally and then diagonally. Working out which features correspond when the chair is cut diagonally is harder than when it is cut horizontally. This example shows that it cannot be assumed that the corresponding features will be found at the same coordinates in adjacent sections after the sections have been registered. Therefore, registering the sections will not necessarily involve placing features in one section directly below or above features in the adjacent sections. Note that "best fit" registration algorithms assume that the distribution of structures in the image is isotropic so that a single structure does not dominate and be reconstructed as a vertical prism rather than its true shape. In addition local "slanting" disturbances are assumed to cancel each other out.

In general however, the sections will be deformed. This poses problems in addition to those described above. There is a problem of distinguishing between biological shape changes and deformations. Both may occur simultaneously. If a feature is bigger or smaller than the corresponding feature in an adjacent section, it could be due to stretching or compression of the section, but it could also be due to the shape of the structure. Compression in the cutting direction may cause features to move relative to each other, but as shown above, this can also occur when structures are not cut perpendicularly to a major axis. Note that large degrees of deformation such as occurs when a limb, or other jointed part articulates, do not have to be taken into account because the embedding medium should prevent such deformation. This implies that we can assume local continuity in the deformations within each section.

An additional problem is determining how much to warp the sections. Care must be taken not to warp sections so that they become identical to an adjacent section. The deformations are rarely obvious when a single section is examined: everything still looks smooth and continuous, but they become obvious when the reconstruction is sliced in a different direction to the original cutting direction [46]. Dürr [26] and Jansson *et al.* [49] correct deformations using information from two sections (the current section and the previous one), but when only two sections are considered, there a real danger of warping one so that it matches the other exactly. For example, if there are no constraints on the amount of warping allowed, a sphere could be reconstructed as a cylinder (figure 2.5).

The problems discussed above must be overcome whether the reconstruction is per-

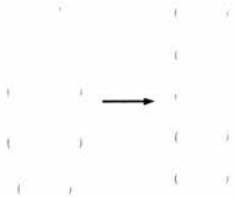


Figure 2.5: If there are no constraints on warping, a sphere could be reconstructed as a cylinder.

formed manually or automatically. If the reconstruction is performed automatically, there is the additional problem of matching adjacent sections. Biological specimens tend to be very complex, and have poor contrast making it hard to pick out the interesting features. Staining tends to be specific to certain structures within the specimen, further limiting the number of features which can be picked out easily[1]. In this work instead of matching features to features we match attributes of "interesting points" to regions in an adjacent image. See chapters 7 and 8 for details.

In this chapter we have discussed the history and the problems of 3D reconstruction from serial sections. In chapters 5 to 10 we will describe and test a method for correcting distortions. This method allows biological shape changes to be conserved by modelling each section as an elastic plate (which restricts the amount of warping), and applying forces calculated by matching each section to *both* adjacent sections. But first we discuss the relevant literature and test some published registration algorithms.

[1] Moss [69] suggests staining the specimen with several different coloured stains, then using colour to pick out the features. He suggests taking pictures in different coloured lights to pick out the different colours.

Chapter 3

Literature Review

In this chapter we will review algorithms for matching images with particular emphasis on registering and warping serial sections. The review has been divided into three sections. The first section reviews methods from the serial section reconstruction literature. Since most methods used to register medical images are also used to register cut serial sections, relevant medical image registration methods which have not been employed in the serial section literature are reviewed along with the serial section registration methods. In the second and third sections we review related literature which includes warping algorithms devised for matching and warping medical images, and image matching and correspondence algorithms employed in the fields of model matching and motion.

3.1 Reconstruction From Serial Sections

There has been research into computer support for three dimensional reconstruction of serial sections since the mid 1960's. In 1986 Huijsmans *et al.* [46] undertook a review of 58 software packages written between 1965 and 1985. Most of the packages written to date rely on substantial human interaction, but there is an increasing number of systems which will automatically reconstruct a 3D object from serial sections.

The main stages in a computer package for three dimensional reconstruction of serial sections are:

1. digitise the image information (grey values or boundaries),

2. identify any fiducials,

3. register the sections,

4. warp the sections, and

5. display the 3D information.

The first stage is to capture in digital form the required information. In most cases this involves defining structure boundaries using, for example, a graphic tablet, or (as in this work) capturing the digital greylevel image. The second stage involves delineating and labelling the features required to align the sections. In most cases these stages are manual and the features may be labelled to establish correspondence with neighbouring sections. For example if an embryo is being reconstructed, the biologist would be able to pick out and label the heart in all the sections containing part of the heart. In some packages, this stage is missed out and the reconstruction starts with registration and warping of the images of the sections.

The registration and warping stages involve bringing the digital images of the sections into correspondence with each other. It may or may not involve deciding which features in one section correspond to features in adjacent sections. In most packages only registration is performed. If fiducials are used, registration is achieved by aligning the fiducials. This is often done manually but in some cases the fiducials can be identified and aligned automatically [97, 98]; in others the fiducials are identified by the user and the computer calculates the required transformation. If there are no fiducials, registration proceeds using a "best fit" method [47].

There are several ways of displaying the reconstruction. If, for example, the connections between neurons are being investigated, a branch diagram may be appropriate however, by far the most common method is to display 3D surfaces. To generate the surfaces it is essential that the boundaries of the various parts of the object are found on each section, and that the correspondences between the boundaries on adjacent sections are known. Surface display facilities can involve hidden surface removal, shading to give depth cues, and perspective projection. Editing facilities can also be included: surfaces may be removed or made transparent; windows may be defined to enable the user

to look inside structures; the user may be able unfold curved surfaces [11]; and the reconstruction may be rotated.

Since the scope of this project concerns registration and warping, only these two stages will be reviewed. Nearly everyone just registers the sections; very few authors have attempted to take deformations into account. Some researchers [84] [38] think that registration is best done by hand as it is then more accurate. Automatic registration is, for them, impossible. On the other hand, manual registration and warping is very tedious and time consuming so some effort has been made to automate it. There are now several methods for automating the registration [42, 2, 1, 85, 99], but warping has proved more problematic, and there have only been a few attempts at fully automating the warping process [100, 49].

There are four main methods of automatically registering and warping serial sections:

- use specifically included fiducial markers;
- use features other than fiducials;
- take an aligned digital image of the section just before it is cut (*episcopic* image) and use this to guide the reconstruction of the cut (and stained) sections (*diascopic* images);
- use the images directly.

The literature will be surveyed under these headings, with extra sections to discuss warping algorithms and methods of feature detection where these have not been covered in other sections.

3.1.1 Fiducials

Fiducial markers are used in some systems but not in others, and various different fiducials are used in the literature. Most authors drill holes through the block [79, 80, 58], and these holes are sometimes filled with other matter[1] [101]. Others embed

[1] The purpose of embedding material in the fiducial holes is in order to be able to stain the fiducials for easy recognition.

the fiducials along with the specimen [97, 98], or use the sides of the embedding block [47, 66].

The main argument for the use of fiducial markers is that registration using fiducials is not subjective, and thereby avoids systematic errors [47]. When sections of a lung, for example, are registered without fiducials, there is often a noticeable sideways shift. When the best fit method is used, slow curvature changes tend to be suppressed [47]. Also, as shown in chapter 2, if each section consists of a circle, it is possible to reconstruct the object either as a cylinder or a spring (or some combination), and if a sphere has been sectioned, it is possible to reconstruct it as a cylinder if there are no constraints on the warping.

Fiducials provide a straight forward mechanism to reconstruct the embedded specimen but are limited in accuracy. They do not always give good results if they can't be used to correct deformations. To correct the deformation, it is necessary to know the exact relative positions of the fiducials so that the tissue can be stretched out accordingly. There must also be enough fiducials as deformations tend to be local. If there are not enough, some deformations will be missed, and other deformations will be introduced as the tissue is stretched uniformly between the fiducials. Clearly, to identify all the deformations using only fiducials would require so many that there would be little or no tissue left.

For the purposes of this project the use of fiducials was not considered because reconstructions from existing section series were required, and fiducials are not available for these sections. We also wanted our method to be as widely applicable as possible and not restricted to the existence of certain types of fiducials.

3.1.2 Registering Features other than Fiducials

Sun [95, 96] and Merickel [65] both use a number of points derived from the images to register the sections using a least squares fit. Zhang [105, 106] uses a pattern matching method based on point constellations to find corresponding features, the features are then registered individually. White and Perednia [102, 74] register images using one to three initial point correspondences, either chosen manually or calculated by a graph

matching technique [102]. The points are extracted manually. If three initial correspondences are given, the images are warped as one triangle is mapped onto the other; two correspondences allow a scaling correction; and if only a single corresponding point is given a translation is performed to align the two points and a rotational correction is estimated by calculating a match measure for a series of rotations.

Least Squares Fit of a Number of Correspondences

Sun's sections consist of coal dust particles. For the method to work, there must be at least three objects in each section. The accuracy of the registration process increases with more objects. There are two stages to the registration process: a coarse registration and a fine registration. Both stages rely on finding and aligning the centres of mass of the objects. The coarse stage relies on the operator choosing three objects in each of the two sections which correspond. The computer checks that the objects have similar shape. If this criterion is met, the match is accepted otherwise it is rejected. When all three matches have been accepted, a registration transformation (rotation and translation only) is calculated to align the centres of mass of the three chosen objects as closely as possible.

The shape similarity measure which is used consists of arranging the two shapes so that they are each contained in a square of the same size and their long axes correspond to the same diagonals in the squares. The intersection operation is performed for the two squares and the resulting area is denoted by C. The shape similarity, S, is[2]

$$S = \frac{C}{A + B - C}$$

where A and B are the areas of the two objects contained in the squares.

The fine registration stage is fully automatic. The computer searches for matching pairs of objects — one from each section. As a rough registration has already been performed, given a feature in one section, the search can be restricted to a local neighbourhood in the other section. All candidate matches are checked using the similarity of shape criterion. The computer then aligns the centres of mass of all the objects which have

[2] In his thesis, Sun gives the similarity measure as $S = \frac{C}{A+B+C}$ which must be a typographical error because it does not have the stated behaviour.

been matched successfully. Only rotation and translation are used, so some trade-off is necessary when it comes to matching several objects because the centres of mass will generally not match exactly. This method was found to be stable for sections involving 20 to 30 objects in that, if a few of the matches are wrong, the overall result was not changed very much.

Merickel's [65] registration algorithm is very similar to Sun's algorithm in that the least squares fit is used to calculate the required transformation from a number of pairs of corresponding points. The algorithm was evaluated using two different types of points: hand-picked contour points, and shape points which were calculated automatically from contours — the centroid and the radius weighted mean point. At least three points are needed for the algorithm so the shape points were calculated for a number of contours. The algorithm, was tested on a simple closed curve (using hand picked contour points only) and on sections of a chick brain. The contours for the chick brain were found interactively and labeled.

The results were evaluated using the mean squared error to measure how well the selected points match and a similarity measure to measure the how well the contours match. The similarity measure was the quotient of the number of similar points in both contours and the number of points evaluated in both contours. Similar points were found by traversing each contour while examining a neighbourhood for points of the other contour.

Good results were achieved with both methods of choosing points giving about 90% similarity along the contours for a 7×7 neighbourhood. The mean squared error was minimum for the shape points.

Matching "Constellations" of points

Zhang [105, 106] considers sections which consist of a lot of megakaryocyte cells obtained from bone marrow. The cells are extracted from the images using a thresholding algorithm and the user selects some of the cells for the computer to reconstruct. Registration involves finding the corresponding cell boundaries between the sections and then approximating each cell boundary as an ellipse so that the boundaries belonging

to each cell can easily be registered by superimposing the centre and axes of the ellipses.

To find the corresponding cell boundaries, each boundary is approximated by one point: the centroid. The section which is used as a reference is called the matched section, the other section is called the matching section. Similar terminology applies to the points in each section. A pattern is constructed for all the selected points in the matched section. The pattern is made up of the selected point and its m nearest neighbouring points. It is specified by the m distances, d_i, from the m surrounding points to the centre point and the m angles, θ_i, between pairs of distance lines. This information is represented by a vector:

$$P_m^l = (d_i, \theta_i, ..., d_m, \theta_m)$$

where l is the label of the centre point. The longest distance gives the radius of the pattern for the matched point. Patterns are also constructed for all the candidate matching points. These patterns have the same radius as the pattern for the matched point but the number of surrounding points may be different. This happens when one or more of the surrounding points in the matched pattern do not have corresponding points in the matching pattern.

To work out which patterns correspond, two types of test are performed. The merit function is a function of the distance between a point in the matched pattern and the nearest point in the matching pattern when the two patterns are superimposed. The similarity function is the root mean square of a set of merit function calculations. It is used both to determine the orientation which gives the best match between two patterns; and to determine which matching pattern corresponds to the matched pattern.

The centre point of the matching pattern is superimposed on the centre point of the matched pattern. The matching pattern is rotated until one of the distance lines coincides with a distance line of the matched pattern. Then the merit function is computed for all the surrounding points in the matched pattern. This is repeated for each configuration of the patterns obtained by rotating the matching pattern. The best orientation is given by the configuration for which the similarity function gives the highest value. The above stages are repeated for each candidate matching pattern and the best match is judged to be the pattern which gives the highest similarity value.

Large distances will generally occur for those points which do not have corresponding points in the matching pattern. These distances may make the final value invalid, so large distances are ignored. The actual merit function used by Zhang is based on the normal distribution function and is inversely proportional to the distance between a pair of compared points.

Discussion

With the exception of the method developed by Merickel, and possibly that given by White and Perednia who do not specify how the points can be found, the above methods are designed exclusively for sections containing several discrete objects where the correspondences between objects in different sections have to be found. Sun does this by using a measure of shape similarity; Zhang matches constellations of points. Aligning the centroids of objects assumes that the object is orientated perpendicularly to the cutting direction. In the case of a single sectioned object this is a valid assumption, but if the sections consist of more than one object, the objects may be orientated in any direction. In this case, if the centroids are used to register the sections, and if they do not fit exactly, it should not be assumed that this is due to deformations in the sections. Merickel's method is designed for sections containing a single object. The problem with this method is that it relies on contour extraction which is very difficult to automate for serial sections in general. Similarly the methods described by White and Perednia are unsuitable for automatic systems because of the difficulties of extracting the same points on two adjacent sections.

3.1.3 Episcopic and Diascopic Images

Lamers and Laan *et al.* [60, 59] suggest taking a digital image of the section before it is cut the *episcopic* image. This produces two sets of sections: a set which is perfectly registered and has no deformations, but has low contrast, little specific staining, and has some projection of underlying structures; and a set which has good selective staining and good contrast, but is unregistered and deformed. The image of the cut and stained section is the *diascopic* image.

The user picks three corresponding points between the two sets so that the computer can register the cut sections and correct the deformations. Registration and warping are performed simultaneously using a global function based on the three selected points. The diascopic triangle is mapped onto the episcopic triangle by a linear transformation. After this, the images are still not perfectly registered and there may still be some deformations, so there is a further registration process which assumes that the centroids of the sections do not move very much. The centroids are brought into line via a weighted averaging of the positions of the centroids of the two adjacent sections.

Lamers and Laan *et al.* suggest that using more points for the registration and warping would allow better algorithms to be used for warping, but this has not been implemented due to the difficulty of picking out more than three suitable points. They suggest either a least squares fit using the existing global algorithm, or a piecewise linear transformation using the existing algorithm on parts of the image. Some non-linear algorithms have been tested but were found to give worse results.

Verbeek [100] has extended the work of Lamers and Laan *et al.*. He automatically calculates the outer contours of the episcopic images and the inner and outer contours of the diascopic images. The outer contours of the two types of images are matched using major concavities. This provides a global transformation. Then the outer contours of the diascopic images are warped onto the outer contours of the episcopic images using a method similar to that described by Burr [17] [18] and Moshfeghi [68]. (These methods are described in detail below.) Distance transforms are used to calculate the feature displacement vectors. Once the warp function for the outer contour has been calculated, the displacement vectors are interpolated over the whole image.

In principle this technique is the ideal technique for reconstructing the embedded specimen exactly and Verbeek shows that some feature extraction and warping can be performed automatically. The problems are that the information extracted from the episcopic image may be limited by poor contrast and artefacts arising from structures from other sections. These methods are of no use for historical data and would be very difficult to use on long series of embedded sections[3].

[3] Serial sections cut from araldite embedded material are technically very difficult to obtain and the additional problem of image capture for each section would make it virtually impossible to achieve a complete series.

3.1.4 Registering Images Directly

There are several methods for automatically registering serial sections directly. The most popular match measure is cross-correlation, but other measures have been developed. All the methods reviewed here work well for the applications on which they were tested, and produce reconstructions of a similar quality (using visual criteria). However since there have been no studies aimed at comparing all the various methods, it is difficult to compare the results objectively for the purpose of registering mouse embryo sections.

Registration using Cross-Correlation

Hibbard *et al.* [41, 43] automatically register edge-enhanced images of sections of a brain. The registration is done in two stages: coarse registration consists of finding and aligning the centres of mass and the principal axes of the object in each section; fine registration involves cross-correlation. There are two parts to the fine registration algorithm. To fine-tune the translation, the cross-correlation of two images is maximised. To fine-tune the rotation, the images are first resampled with polar coordinates and then the cross-correlation of these new images is maximised. These two steps are alternated until nothing changes. This is, in fact, done at two resolutions. The images are first polar resampled over 360 degrees at 2 degree increments. When an optimal fit is achieved at this resolution, the images are polar resampled over 90 degrees at 0.5 degree increments. The 90 degree region selected should include small high contrast features which are far from the image centre.

Hibbard *et al.* use images edge enhanced by the Robert's cross operator for the cross-correlation stage as it was found that this gave more reliable results than when the original images were used. The cross-correlation stage is designed to correct the errors made by the first registration stage. Aligning the centres of mass of each image may not necessarily give a good registration. For example, if a significant branching structure appears on one side of the object, the centre of mass will move. The top section is fixed and all the other sections are sequentially transformed to the previous section. This can propagate errors in the registration procedure. That is, a sideways shift

in an early section will cause the shift (with respect to the top slice) to be present for the rest of the reconstruction. If the registration fails (when the principal axes alignment fails, for example), the solution is to manually rotate the sections to within 90 degrees. This method still works if the sections are slightly damaged but fails when the sections are very damaged. In particular the system fails when the features are abruptly clipped or when features suddenly appear or disappear. Hibbard *et al.* make no attempt at correcting any deformations. To speed up the calculations, cross-correlation is calculated using the fast Fourier transform.

To correct some of the problems with the earlier method Hibbard *et al.* have subsequently developed a new method [40, 42]. This cross-correlates the polar resampled autocorrelations of the images, instead of the images directly, to estimate the rotation correction. This dispenses with the centroid and principal axes alignment as the autocorrelations are independent of translation. When the images are sufficiently well aligned, fine registration proceeds by cross-correlating the polar resampled images as before. In addition the images are smoothed by a low pass filter before cross-correlating to estimate the translational correction. This new method was found to reduce the number of failures.

Apicella *et al.* [2], who register images from different modalities, separate the rotational and translational components of the registration by using the power spectra of the images. They also correct any difference in scale between the images. The registration is in three stages. First the scale, then the rotation and finally the translation are found and corrected. Before the rotation can be corrected, the images must be modified so that they are translation independent. This is achieved by taking the power spectra of the images. The rotation is found by polar resampling the power spectra and cross-correlating. The scale cannot be corrected until the images have been rendered translation and rotation independent. The rotation in the polar resampled power spectra is represented by a translation so the power spectra of the polar resampled images will give images which are invariant to both rotation and translation. The multiplicative scale component can be represented by a translation by performing a logarithmic transformation of the spatial domain coordinate system to convert the scale component to a shift.

This method was found to work on test images but initial tests on real images (PET images reduced to the size 64×64) produced unacceptably large errors because the real image was rotated within a square window with background noise. The problem was that taking the power spectrum effectively convolved the image with the square window containing the image. When the image is rotated the square window does not rotate with it and therefore the power spectrum of the rotated image is not a rotated version of the power spectrum of the original image. To remedy this a Hanning window was applied to the image before calculating the power spectrum. This has the effect of multiplying the image by a function which has value 1 in the centre of the image, 0 at the edges and varies smoothly in between. Applying the Hanning window to the image had the desired effect of removing the unwanted spectral components, and reducing rotation errors.

Toga and Banerjee [99] have also developed a method based on cross-correlation. After a rough registration using centroid and principal axes alignment, this method calculates correspondences by matching small areas using cross-correlation. The correspondences are weighted by considering the best positions of the four neighbouring grid points so that spurious matches are given less weight, then the transformation is calculated using a least squares minimisation technique. This is repeated until there is no change in the transformation. The authors compared this method to the most recent method developed by Hibbard *et al.* [40, 42] and found that the two methods gave similar results.

Other Registration Methods

Andreason *et al.* [1] calculate a measure of fit based on the sum of differences of a number of pixels. 1000 pixels are chosen at random from one image and these are matched to the pixels in the same position in the other image. Then one image is rotated or translated and the measure is recalculated. The actual measure used is:

$$\Delta = \left[\sum_{i}^{n} \left(\frac{I_{1_i} - I_{2_i}}{C_i} \right)^2 \right]^{-1},$$

where I_{1_i} and I_{2_i} are pixels in corresponding positions in image 1 and image 2 respectively, n is the number of chosen pixels, and C_i is the larger of the ratios I_{1_i}/I_{2_i} and

$I_2,/I_1,$. To find the best match for each of x, y, and θ, the values of the match function are plotted for each of a series of angles and displacements in x or y. The value of x, y, or θ giving the peak of the match function is estimated by dividing the area under the curve, defined by the appropriate set of plotted points, into two equal areas and averaging the two indices which give half the maximum value in their respective areas. The above is repeated until there is no change. This still does not produce a satisfactory registration due to accumulative errors. To correct for this, instead of defining the total transformation for each section to be the sum of the transformation of the current section with respect to the previous section and the transformations of all the previous sections, it is defined to be a weighted sum of the transformation of the current section with respect to the previous section and the transformations of the previous n sections.

Rydmark *et al.* [85] register thresholded images. The images are thresholded at 127 (half the maximum grey value) and the index of mismatch is defined as the relative number of zero pixels in the image obtained by applying the exclusive OR operation to the two images. This index is calculated as one image is rotated and translated with respect to the other, and the best transformation is defined as the one which gives the lowest mismatch index. The above is repeated, successively reducing the size of the rotations and translations, until lower mismatch indices are no longer obtained. This algorithm is applied after the sections have been roughly aligned manually.

Herbin *et al.* [39] have devised an algorithm for registering medical images of the same patient taken at different times. The algorithm is specifically designed to cope with differences in the images. Herbin's method is to maximise a similarity criterion using the Adaptive Random Search (ARS) optimisation algorithm. Three possible similarity measures are suggested: the stochastic sign change criterion (SSC), the deterministic sign change criterion (DSC), and the outlier minimal number estimator (OMNE). The SSC is recommended for noisy images and is defined as the number of sign changes in the subtraction image

$$D(i,j) = F_2(i,j) - F_1(i,j)$$

scanned line by line *or* column by column, where F_1 and F_2 are the two images which are being compared.

For images where the noise level is low compared with the precision of the digitisation,

the use of either the DSC or OMNE is recommended. The DSC is defined as the number of sign changes in the subtraction image

$$D(i,j) = F_2(i,j) - F_1(i,j) - q(-1)^{i+j}$$

scanned line by line or column by column, where q is a small real or integer value. The OMNE is defined as the number of greylevel values of $F_2(i,j)$ which belong to the associated interval $[F_1(i,j) - q, F_1(i,j) + q]$ where q is a small integer value.

The main idea of the ARS strategy is to compute the chosen criterion for randomly chosen parameter values (rotation and translation, for example). When the criterion value is close to its global maximum, only a small region in the parameter space should be searched and when the criterion value is not close to the global maximum, a larger region should be searched. To achieve this, sets of variances are chosen and new trial points are generated so that each new trial parameter forms a normal distribution with a particular variance. Each set of variances contains a variance for each parameter and the sets are ordered so that each successive set of variances contains a smaller variance for each parameter. The algorithm consists of three stages: initialisation, choosing a variance, and using the chosen set of variances in an attempt to randomly generate new trial points which give a higher criterion value. The last two stages are performed alternatively until either the set of smallest variances has been chosen a fixed number of times consecutively, in which case it is assumed that the global maximum has been found, or until a fixed number of iterations have been performed.

Discussion

All the registration algorithms described above work for the applications for which they were designed. With the exception of Hibbard's method they should all work on images where the object is not fully contained in the image. Hibbard's method will not work in this case because the Fourier spectrum will be convolved with the square image. This problem can be solved by applying a Hanning (or other) window to the image before calculating the Fourier transform so that edge effects are reduced, as described by Apicella *et al.*. The drawback with this is that the contribution of features far from the centre of the image is reduced although these are the best for determining the error

in rotation. Note that Hibbard's later method is similar to using Apicella's method to calculate an initial rotation. Hibbard's method may produce better results than Apicella's method because of the fine registration stage.

By inspection of the results of the various registration algorithms contained in the literature, it was found that the methods all produce results of a similar quality. All the algorithms have similar problems in coping with certain types of images. For instance, all may fail if consecutive images are significantly different, such as when an object suddenly becomes much larger or much smaller. These types of images will require manual correction. In addition all methods, with the possible exception of that given by Toga and Banerjee, will have difficulty in registering concentric rings of significantly different radii as they all assume that the best registration position is where the pixel values match best. This is not necessarily true for images which are different. However for most applications the images are similar enough that this approach gives satisfactory results. The fact that the above methods seem to give similar results may indicate that the remaining misalignment is due to deformation in the sections.

3.1.5 Deformation Correction

Very little reported work mentions the local deformations which arise from the preparation of the specimen. Even when fiducials are used, if there is no correction of the local deformations, there will be a visually significant degree of misalignment. Various methods of warping have been described, but most require a certain amount of interaction. Prothero *et al.* [80] and Baba *et al.* [4] correct deformations manually: the user is able to move individual contours and even manually redraw all or part of the contour. Yaegashi *et al.* [104] measure the dimensions of the sections as they are cut. It was found, in this study, that as each successive section was cut, the deformation increased linearly until the knife was renewed. This deformation is corrected by a linear expansion or reduction in the x and y axes respectively.

The main methods of distortion correction are surface smoothing, global warp transformations, and elastic warping. Both global and elastic warping methods aim to warp one set of landmark, or tie points onto another. *Landmark*, or *tie points* are pairs of points which which denote correspondences between sections.

Surface Smoothing

Deverell *et al.* [24] avoid the problem of warping by removing contour points until the number of points is small enough to give a smooth reconstruction. Gérard *et al.* [35] take a similar but more sophisticated approach. They use an algorithm which smooths the surfaces of the reconstruction without removing the significant details. This is achieved by the use of the DSI (Discrete Smooth Interpolation) algorithm which was designed for easy modelling of complex 3D geological surfaces. The algorithm interpolates the surfaces by selecting points on the contours to retain. These points are selected according to the curvature radius so that the more complex the contour, the more points are retained. The user can interactively try different levels of smoothing to get the best final display. The method described by Gérard *et al.* requires that the images are registered and that contours are extracted and the correspondences between the contours are known. It is, in effect, a sophisticated surface generation algorithm rather than a warping algorithm. Another surface smoothing approach is described by Jansson *et al.* [49] who truncate Fourier descriptors of (manually traced) contours. Again, the degree of smoothing can be controlled by the user.

Global Warp Transformations

Lamers and Laan *et al.* [60, 59] and Kriete et al [58] use a global transformation algorithm based on three points. As an extension of this, Olivo *et al.* [72] warp the sections using polynomial functions so that user defined points are brought into alignment. A polynomial of degree 1 is used when only scaling and shear are required; a polynomial of degree 2 is used when global non-linear warps are required. If local deformations are present a polynomial of degree 1 is used and this is followed by a non-linear local transformation. This transformation essentially calculates the displacement for each pixel by averaging the displacements of the tie points weighted by their distance from the pixel.

Jansson *et al.* [49] use global linear functions to warp their sections. However in this case adjacent images are matched and a displacement map is produced automatically. After the sections have been registered using the method described by Rydmark *et al.* [85],

the deformation correction procedure uses the index of mismatch described by Rydmark *et al.* to match neighbourhoods of both images. The offset giving the minimum value is the displacement vector. The displacement vectors are smoothed by applying median filters independently to each coordinate, and a linear warp function, including expansion and compression only (the sections are assumed to be registered), is calculated. The direction of the warp is calculated by choosing the most common direction of the vectors. To calculate the magnitude, the x and y displacement components are plotted against their x or y (respectively) position. The slope of the regression line fitted to this data gives the magnitude. This method seems to correct most of the deformation of their plastic serial light micrographs, which when analysed for deformation [10] were found to contain deformations mostly due to compression in the cutting direction and expansion perpendicular to the cutting direction. Electron micrographs were found to contain more non-linear deformations.

Modelling Sections as Elastic Plates

An alternative approach to warping based on tie points is to model the sections as thin elastic plates. This is the approach taken by Bookstein [13, 14] and Dürr [26]. Dürr *et al.* [26] use an elastic plate model to correct the deformations in the sections. The idea is to minimise the deformation energy

$$E_D = \frac{1}{2} \iint \boldsymbol{\sigma}^T \boldsymbol{\epsilon} \, dx \, dy$$

where $\boldsymbol{\sigma}$ is the stress vector and $\boldsymbol{\epsilon}$ is the strain vector. The user has to specify the required movements of some arbitrary points (passpoints) in the image. These passpoints serve as boundary conditions for the elastic plate model. The accuracy of the method can be improved by choosing more passpoints. A displacement field is calculated by numerically solving the equation system above, and used to correct the image.

Bookstein uses thin plate splines to map one set of points onto another while minimising the bending energy

$$E_B = \iint \left[\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 + \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 \right] dx \, dy,$$

where f is the function which maps one set of points onto the other. Note that in this method the tie points are not necessarily fixed, but can be allowed to move along

a tangent line. This gets round the problem of needing to locate all the tie points precisely.

Discussion

Methods which use global linear or polynomial transformations assume that the sections have been uniformly deformed. This is not always the case because although there is usually some systematic deformation, wax sections, in particular, are likely to contain significant random deformations. Methods which use small numbers of tie points are likely to be unsatisfactory because of error magnification in regions which are far away from the landmark points. If the landmark points are chosen close to the centre of the section, there will be large warping errors at the edges. Using a piecewise linear transformation based on more points may improve the results. This would ideally involve triangulating the image and performing a linear transformation in each triangle, making sure that the boundary conditions on the edges of the triangles are compatible, as suggested by Laan *et al.* [59]. The problem with this is that it is difficult for the user to reliably indicate a lot of tie points as the tie points are restricted to special, easily identifiable points of which there are generally few in the image.

So far only Dürr, Bookstein and Verbeek have considered local non-linear transformations. Dürr and Bookstein have applied the theory of elasticity to the problem of warping serial sections, but they have proposed no methods of automatically finding correspondences. Verbeek's method is fully automatic but requires that an outer contour can be extracted from both the episcopic and diascopic images. These methods all warp one set of tie points or contours onto another so that one image becomes identical to another image. This is ideal behaviour for problems where an image is mapped onto a control image or an atlas, but it is unsuitable for the problem of warping serial section images. To correct for local deformation in serial sections, shape change effects must be separated from deformation effects to prevent sections being warped so that they become identical to an adjacent single section. This requires that both adjacent sections, at least, are taken into consideration when calculating the warp. None of the methods described above tackle this problem, and are therefore unsuitable for correcting local deformations in serial sections except when episcopic images are employed.

3.1.6 Extracting Features

Matching images as a whole may be adequate for solving the registration problem, but is of no use for solving the warping problem. Since warping involves correcting local deformation, it is essential that the images are matched locally. There are two approaches to local image matching: the first is to match features which have been extracted, and the other is to match the pixel values in regions of one image to the pixel values in regions of another image directly. Cross-correlation is most often used for the latter. Here we review the image processing methods used for feature extraction in serial section reconstruction.

Automatic feature extraction is very difficult for biological images. In many packages the user has to trace and label the contours in such a way that the computer knows which contours belong to which objects in all the sections. Sometimes the computer is expected to recognise some of the features [75, 41]. Obviously, if fiducials are included, they should be picked out and recognised. Fiducials are generally picked out by hand. Toga *et al.* [97, 98] have automated the process by thresholding the images, which produces three main objects: two small objects and one big object for each image. The fiducials are the small objects and are found by counting the number of pixels. To avoid confusion with artefacts, there is a lower limit on the number of pixels which can comprise a fiducial. Moss *et al.* [69] simplify very complicated images using erosion and dilation. They find the simplification necessary in order to interpret the final reconstruction.

Hibbard *et al.* [41] detect capillaries in the median eminence (part of the brain). They searched for mathematical properties which would distinguish the capillaries from the rest of the features in the sections. It was found that capillaries can be detected by their high greylevel values, their low local variance and the fact that they are the biggest connected features in the images. Therefore, the capillaries are detected by thresholding on the grey value of the pixels, and on the standard deviation of the greylevels of the pixels and their eight nearest neighbours.

Perkins and Jordan [52, 75] argue that an expert system is needed for feature extraction and recognition in biological images. Conventional image processing is not good enough

as the features may be distinct in some cases but not distinct in others. To find and identify features, expert biological knowledge is needed. The expert system has not been written yet but the authors have set out in some detail how they intend to solve their problem. The features found by conventional image processing techniques will have to be matched to a model of the object which is to be reconstructed. Biological knowledge will be essential and will include information about the positions of features relative to other features, their position in the series of sections and their shape relative to the position of the section in the series. As some of the features will be modelled relative to others, it will be important to search for them in an appropriate order. For example, if feature B is modelled relative to feature A, there is no point searching for it until feature A has been identified. Features will be identified using information about shape, texture and density. The option of user interaction will be included to train the system and to correct any errors.

From the above we see that although there are specialised algorithms for automatically extracting specific features in specific types of sections, there are no general feature extraction methods.

3.2 Matching Tomographic Images

There are two areas of related work: multimodality matching which involves matching images from the same person but using different imaging techniques (CT and MRI scans, for example) and matching a CT scan of one person to a CT scan of the same, or another, person, or a model.

Multimodality matching may involve registration and warping because different scans can involve scanning the patient in different positions and different imaging devices may deform the images in different ways. Another problem is that, in general, the grey values will not correlate. When scans of different people but of the same modality are matched, the sections are deformed with respect to each other because of the individual variations in anatomy between people.

The problems posed by these areas of work are very similar to the problems of matching diascopic images to episcopic images, discussed above. The sections have to be warped

so that they match the reference image exactly and the same features are expected to be present in both sections. The methods used to solve these problems include landmark matching and elastic matching. Landmark matching involves manually picking out salient points (landmarks) in the images and then warping one image onto the other using polynomials. More recently, various other forms of matching have been proposed: Bajcsy *et al.* [5] use an elastic model to warp the sections, Mowforth *et al.* [70] and Woodward [103] produce a map which maps all the pixels in one image to the corresponding pixels in the other image, and Moshfeghi [68] warps the contours in one image onto the contours in the other image.

Methods Based on Cross-Correlation

Bajcsy *et al.* [5] match CT scans of the brain to a model of a human brain derived from an anatomy atlas. The matching is in two stages: a global registration stage and a fine elastic warping stage. The global registration stage aligns the centroid and the principal axes of the sections; the fine warping stage uses an elastic model to warp a CT scan section to the corresponding section in the atlas data. Warping is performed at several resolutions: first the main features are matched, and then the finer details are fitted as the resolution increases.

Bajcsy *et al.*'s elastic model assumes that the deformations are small. Their method involves applying external forces to the elastic plate to achieve the required warp. The amount of warping allowed is limited by the elastic constant μ. When μ is large, the plate is more rigid. When μ is small, the solution is controlled mostly by the applied forces, also any noise will be more obvious and false matches are more likely. So a moderately large value for μ should be chosen. To find the forces to apply to the plate, a similarity function based on cross-correlation is used. The idea is to increase the similarity function by applying a force proportional to the gradient vector of the similarity function. As only local deformations are required, the similarity function is only applied to small areas of the image. The algorithm calculates the similarity function over small regions in the image, calculates the forces required to warp the image, and then warps the image. This is repeated 16 times for each resolution. The number of iterations was determined empirically and is fixed so that a stopping condition does not

need to be considered.

Mowforth *et al.* [70] match MRI images of the brain to an MRI model of the human brain derived from an anatomy atlas. Their matching process also involves two stages: a global matching stage and a fine elastic matching stage. Global matching involves manually registering and stretching the MRI image to bring it into rough correspondence with the model; fine elastic matching involves finding a displacement map between the pixels in the MRI image and the pixels in the model. Matching is performed at several resolutions in a coarse to fine progression. Given a certain discrepancy, cross-correlation is used to search the neighbourhood of the pixel which is expected to correspond to a given pixel in the model. When an optimal match is found, a new discrepancy is produced and fed to the next resolution. The initial discrepancy is assumed to be zero. The discrepancy produced at the finest resolution gives the estimated discrepancy maps. Using these discrepancy maps, it is possible to identify all the parts of the brain which are included in the models and their areas.

Woodward [103] is interested in finding the changes between two MRI scans of the same person taken at different times. His method involves finding point to point correspondences between the two scans, smoothing the resulting displacement vectors and then finding residual displacement vectors after the global transformation has been extracted. The two images are roughly registered manually. Point to point correspondences are found by cross-correlating small regions around pixels in the images. As the images are already roughly registered, the search for the corresponding pixel can be restricted to a limited area in the second image. The resulting vector field is smoothed iteratively using weights assigned to each vector, where the assigned weights are the normalised product of the cross-correlation and the standard deviation of the greylevels around each pixel. This gives pixels which lie on edges more weight, which implies that the correspondences along edges are expected to be more accurate. The global transformation is extracted by a least squares method, and the remaining displacement vectors indicate the changes to the MRI scan.

Contour Matching

Moshfeghi [68] matches CT scans to MRI scans. The algorithm is based on Burr's [17] [18] method of deforming one curve or contour into another. Moshfeghi extracts the contours manually; correspondences between contours are also found manually. The contours are made up of short line segments. For the algorithm to work, the sections must be roughly aligned and of similar size.

Moshfeghi's algorithm consists of the following steps:

1. Roughly register each contour (align centroids).
2. Calculate the displacement vectors.
3. Calculate, and apply the force field.
4. Repeat 2 and 3 until the magnitude of the displacement vectors are below a given threshold.

Displacement vectors are calculated for each pair of contours using a similarity measure based on the direction of the corresponding line segment and on the shortest distance from a point on the reference contour to the line segment on the corresponding contour. These vectors are used to calculate the force field which is used to warp the sections. The force vectors for each contour are calculated by taking the weighted average (Gaussian) of the displacement vectors over a certain distance. The neighbourhood is large for the first few iterations but it gradually decreases. The sections are effectively modelled as stiff plates when the neighbourhood is large because displacing a single point will have effects over a relatively large area. When the neighbourhood is small, the behaviour of the sections is less rigid and small details can be brought into alignment. The force field is calculated for all the contours and then interpolated over the whole image.

Gradient Matching

Goshtasby *et al.* [34] combine intensity and gradient information to calculate correspondences between adjacent tomographic sections in order to interpolate new sections.

Only points with high gradient magnitudes are matched, and the best match within a local search neighbourhood is chosen as the correspondence. To avoid false matches, matching is performed in both directions and the displacement maps are smoothed. Only consistent correspondences are retained; the displacements of other points are linearly interpolated.

Discussion

With the exception of that proposed by Goshtasby *et al.*, techniques for matching CT and MRI scans cannot be applied directly to the problem of warping serial sections because, in this application, some features in one section may not have corresponding features in adjacent sections. When CT or MRI scans are matched, the assumption is that the differences will be small. Therefore the techniques are not designed to cope with major changes in the images such as features appearing and disappearing and branching structures, where one feature in one image corresponds to two, or more, features in the next image. In these cases, finding correspondences for all the points in the images using cross-correlation will not work as some points should not correlate well with any of the points in the adjacent image. If this method was used to warp serial section images, it would be essential that the system can detect cases where points do not have corresponding points. The method proposed by Goshtasby *et al.* does not have this problem because points without correspondences are detected and the displacements of these points are interpolated from the displacements of points with correspondences.

The elastic matching which has been implemented so far, assumes that the bodies can be considered to be homogeneous and isotropic, and cannot easily be adapted to cope with inhomogeneous materials. Nevertheless elastic matching methods are useful for correcting local deformations.

3.3 Model Matching, Motion and Correspondence

We now turn our attention to problems in the field of machine vision where the aim is to interpret images of the real world. In particular we will consider model matching, motion, and general correspondence algorithms. Model matching involves selecting

a model and seeking an instance of the model in the image. There are two main problems: the selection of the model and the location of the instance of the model in the scene. Here we will restrict our attention to the location problem. Motion involves tracking objects in a series of images and calculating their motion with respect to the motion of the viewer. There are two main approaches to this problem: optic flow and feature based methods. Optic flow involves matching the greylevels of the two images to obtain a displacement map for all the points in one of the images; feature based matching calculates the motion from relatively few feature correspondences. We will not discuss how the motion is calculated because we cannot assume any correlation in the deformation of adjacent serial sections.

Correspondence algorithms have also been developed for applications where two images taken at different times, and often from different view points, have to be registered in order to detect changes, for example, in remote sensing where two satellite images have to be registered.

Another field which requires the calculation of correspondences is stereo vision where two images of the same scene are taken from two slightly different positions. Finding correspondences between the two images enables three dimensional information to be calculated about the scene. This field was not considered because the correspondence problem can be reduced to a one dimensional problem, and therefore the techniques are unlikely to be useful.

Correspondence algorithms, in general consist of four parts: selection of features, generally points or edges, to be matched, a similarity measure to compare two individual features, a search strategy for finding the best match of a number of features, and a cost function for determining the best match. Matching generally proceeds in two stages: first individual features are matched, then correspondences are calculated by the application of the search strategy and cost function. If the location of the object is not calculated as part of the second stage, it can be calculated from the correspondences. In the case of model matching this can be included in a verification stage where further model to image correspondences can be found by superimposing the model on the image. Since the particular type of features used depends on the application we will concentrate our attention on the second stage.

In this review we group the various correspondence algorithms by the types of features they employ. First we will consider methods which match features (such as edges and regions) and their attributes (such as edge direction and area); subsequently, methods which use points (with no feature attributes) and greylevels will be discussed.

3.3.1 Feature Matching

Clearly the simplest correspondence algorithm would be to match all the reference image (or model) features to all the target image features exhaustively and score all the possible combinations of reference image (or model) and target image features. This strategy is computationally impractical for all but the simplest of models and images and much research effort has been directed at reducing the search space.

If, as in the case of motion, approximate positions of the features can be predicted, the search can be reduced to a relatively small portion of the image. This is the approach taken by Stephens *et al.* [93] who simply find the best match in the predicted region. Instead of predicting the position of features in the image, some researchers predict what the object should look like in the image and search for groups of features in appropriate positions relative to each other. This approach is taken by Brooks and Lowe. Brooks [15, 16] uses models based on generalised cylinders and thus searches for instances of ribbons and ellipses in the image. Matching proceeds by matching graphs, one representing spatial relationships between these features; the other representing spatial relationships between model features. Lowe [63] clusters the features into "perceptual groupings" on the basis of proximity, parallelism and colinearity. One feature may belong to several groupings. This kind of approach cuts down the search space significantly as only groupings are matched and a grouping will not match in many positions in the image and most matches can be rejected outright.

Fisher [29] uses a similar approach. In this surface matching method, model and image surfaces are each combined into an hierarchical database of surfaces and surface clusters. Properties of data surfaces (and clusters of surfaces) are matched to surfaces in the model database and the plausibility and inhibition of matches are calculated and propagated through an associative network. When the network has converged, models with high plausibility are verified against the image data.

An alternative approach is to order the features so that the least common features are searched for and matched first. This is the approach taken by Bolles *et al.* [12] and Huttenlocher *et al.* [48]. Bolles *et al.* grow clusters of matching features from the initial match by using the model to predict distances to other features. Since little is known about the position of the object in the image, several clusters can be grown from the same object and a graph of possible matches is constructed. The largest subgraph so that every node in the subgraph is connected to every other node in the subgraph (the maximal clique) is identified and this gives a hypothesis which can be verified by superimposing the model on the object. Huttenlocher *et al.* take a different approach. They use the initial match to calculate the position of the object in the scene and then the match is scored depending on the number of features which are brought into approximate correspondence. If the match accounts for more than half the model features the match is kept, otherwise it is rejected.

Tree search is a common matching strategy [36, 32, 3, 67]. This involves potentially exhaustive search by successively matching each model feature to each image feature so that a tree structure is formed. To prevent exhaustive search, the tree can be pruned by backtracking when the cost function exceeds a certain threshold. Additionally the features can be ordered so that the more prominent features are considered first, and the search stops when the verification function exceeds a certain threshold.

Finally we consider the problem of calculating the position of the instance of the model, or one image relative to the other. There are two main ways of doing this: some authors calculate the position from the correspondences [12, 29, 36, 15, 63]; others calculate the position from initial correspondences and update the position as more correspondences are assigned [3, 32, 48]. In the former case additional correspondences can be found, and hypotheses can be rejected, by superimposing the model on the image.

3.3.2 Point Matching

We now consider the problem of matching two sets of points. If the two sets are known to be roughly registered, relatively simple techniques such as those described by Perednia *et al.* [74] and Skea [91] can be employed. Skea *et al.* compare triangles formed by sets of three points in the two images. All the sets of three points in image 1 are matched

against all candidate sets in image 2 and the results are added to an accumulator. All the candidate points in image 2 must lie within a predefined tolerance of at least one of the points in image 1. When all legal combinations of triangles have been tested, corresponding points are those given by the highest values in both the rows and columns of the accumulator.

A point matching algorithm which matches "constellations" was described above; another method based on constellations was described by Poole *et al.* [77] who use a Bayesian statistical model to relocate cells on a previously processed slide. Strickland *et al.* have developed a method based on distances and angles between points which can cope with turbulent motion and deformation. To ensure reliability of matches, the procedure is applied in both forward and backward directions, and inconsistent matches are rejected.

An interesting correspondence algorithm is that described by Scott *et al.* [89]. This essentially minimises the squared distances between corresponding points, but the algorithm is elegant in that it involves a single calculation. A pairing matrix is obtained by matrix algebra from a matrix where each element is a function of the distance between two points: one from each image. Then if P is the pairing matrix, P_{ij} indicates the extent of the pairing between point i in image 1 and point j in image 2. If P_{ij} is the greatest element in row i and in column j, the two features are in 1-1 correspondence. Since the algorithm minimises the sum of distances between pairs of features, it can cope well with translation and deformation, but not so well with rotation. To overcome the problem of rotation Shapiro *et al.* [90] modified the algorithm. In this version the feature vectors are the eigenvectors of the proximity matrices of the two images. The proximity matrix is the matrix whose element a_{ij} is a function of the distance between point i and point j in the same image. The eigenvectors are correlated to produce the association matrix Z which reflects the confidence in the matches between the points where zero indicates a perfect match, and two indicates that the two points do not match. This method copes well with translations, rotations, reflections, and scaling. It copes with deformations as long as the shape of the object does not change too much.

Other authors have extended this idea to model matching. Cootes *et al.* [22] and Hill *et al.* [44] extended the idea to include greylevel information. The models are trained on

examples where the operator provides the model points, and the permitted variation of the *modes* of the model are calculated. The modes correspond to the eigenvectors of the proximity matrix given above, except that modes based on the differences in greylevels along a profile through each model point are also calculated. A genetic algorithm is used to predict the global position of the instance of the model in the image; local matching is used to refine the estimate. Local matching involves local searching for better matches to the profiles and updating the position of each model point by separately taking both greylevel and shape information into account. Sclaroff *et al.* [88] modified the algorithm proposed by Shapiro *et al.* by inserting virtual rubbery material between the points and using the finite element method to calculate the resonance frequencies (modes). In this case the shape model is not learnt, but a physical model is assumed and used to derive a model for matching. An extension which copes with large rotations is also presented.

3.3.3 Greylevel Matching

Methods based on cross-correlation have been discussed above and therefore will not be considered here. An important class of greylevel matching techniques are aimed at calculating optic flow and involve combining intensity and smoothness constraints into a functional equation which can then be minimised using finite differences [45] or the finite element method [87]. The assumptions are that the brightness of an object should not change much as it moves, and that the displacement field should vary smoothly almost everywhere (the exception is at boundaries of objects). The functional equation to be minimised is [45]:

$$E_O = \iint \alpha\ 2E_c^2 + E_b^2\, dx\, dy,$$

where α is a weighting factor, E_b is the sum of the rate of change in image brightness in x, y, and t (time), and E_c is a measure of departure from smoothness in the velocity flow $((u, v) = (\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t})$ defined by

$$E_c = \left(\frac{\partial^2 u}{\partial x^2}\right)^2 + \left(\frac{\partial^2 u}{\partial y^2}\right)^2 + \left(\frac{\partial^2 v}{\partial x^2}\right)^2 + \left(\frac{\partial^2 v}{\partial y^2}\right)^2.$$

Bartels *et al.* [7] proposed this approach as a means of studying biological shape change over time. His particular field of interest is moving cells.

3.3.4 Discussion

Some of the above correspondence methods were designed for deformable objects. This includes, in particular, the point matching algorithms. Other methods can also be adapted to cope with deformable objects: for example tree search [67] and optic flow [7].

Feature matching techniques assume that the same features can be extracted from both images. If this is not the case the calculated correspondences are approximate. These techniques are not suitable for matching serial section images because of the difficulty of locating the same features on adjacent sections. An alternative method is to match feature attributes based on greylevels directly to the other image. This is done in the method proposed by Cootes *et al.* and Hill *et al.* [22, 44] where the model consists of a cloud of points (assumed to be located on boundaries) but greylevel profiles are matched. However this method is not suitable because the models have to be trained and this requires manual input of suitable feature points. In addition it is not clear how well this approach would deal with significant changes such as objects appearing and disappearing between sections.

Optic flow methods match greylevels directly, but it is not clear how these methods would cope with significant changes caused by branching structures or objects appearing and disappearing. In addition, because matching is performed by minimising a functional over the whole image, it would be difficult to detect and deal with such changes when they occur.

3.3.5 Snakes

Kass *et al.* [54] proposed the use of *snakes* or "active contour models" for dealing with deformable models. A snake is an energy minimisation spline which is influenced by image forces that pull it towards features such as lines and edges, and which can be guided by external constraints supplied by the user. Snakes have several parameters including stiffness and parameters to determine the type of edge to which it is attracted. Making the snake less stiff (locally) allows it to produce corners; a stiffer snake ensures a smooth curve.

Snakes need to be placed near the desired contour, but once placed they move until an equilibrium position is reached. This equilibrium is designed to lie along a contour and this makes snakes useful for tracking deformable objects between scenes as each scene can be used to predict the position in the next scene. Cohen and Cohen [21] modified the basic snake by including an inflation or deflation force. This enables the snake to expand or contract to the desired contour making it more robust with respect to its starting position. However in this case the snake must know whether it should inflate or deflate. Sullivan *et al.* [94] investigated the use of snakes for segmenting MRI scans. The snake was placed on one section and allowed to move to an equilibrium position on each subsequent section. It was found that as long as the object did not change shape too dramatically, and in particular did not divide, the correct contour was found. Carlbom *et al.* [19] use snakes to interactively segment serial section images. The user can reinitialise the snake, divide it, and join two snakes. However for our study of fully automatic warping, such methods are not useful.

3.4 Conclusion

In this chapter we have reviewed algorithms for matching images with particular emphasis on the problems of registering and warping serial sections. We have found that several algorithms for registering serial sections have been reported in the literature, but that although there are algorithms for calculating global warp transformations there are no algorithms for correcting local deformations which are suitable for our application.

Although the different registration algorithms reviewed all seem to give similar results in the quality of the reconstruction, they were all tested on different types of images. Since no objective comparison of all the different methods has been reported, they should be tested to see how they perform on images from our particular application. This will be discussed in chapter 4.

The main problems to overcome in order to correct local deformations in serial sections are finding reliable correspondences, and preventing images being warped so that they become identical to another image. Because of the difficulties of segmenting features from biological images, of all the matching algorithms reviewed those that are most

suitable are those which do not involve matching features such as points, edges, or regions to similar features in another image. This leaves algorithms based on matching neighbourhoods of greylevels, such as cross-correlation, and algorithms such as that described by Goshtasby *et al.* [34] where attributes of features extracted from one image are matched to another (filtered) image directly.

Three methods of warping images based on local information are presented in the literature: methods based on Burr's method of warping one line onto another, modelling the sections as thin elastic plates, and simply calculating a displacement map. Of these modelling the sections as thin elastic plates is the most suitable for our application because the elastic model is closest to the real world model of deformations and therefore using elastic material should allow point displacements to propagate properly. In particular it automatically ensures that folding is prevented and neighbouring points remain close together. The methods involving warping one contour onto another are unsuitable due to the difficulty of extracting contours automatically. In subsequent chapters we will discuss how sections can be warped by modelling them as elastic plates and applying forces obtained by matching adjacent sections.

Chapter 4

Comparison of Registration Methods

4.1 Introduction

In chapter 3 we saw that several algorithms for automatically registering serial sections have been described, but that few algorithms for correcting local distortions have been presented. Correction of local distortions requires that many correspondences between adjacent sections are found and that the correspondences are distributed evenly over the images. Clearly, if the sections are registered, the search space for finding correspondences can be dramatically reduced because the search can be restricted to a local neighbourhood centred on the point to be matched. In addition, although warping could be made to register the sections, it is computationally more expensive than registration and therefore if effective registration algorithms exist, their use should be required as a preliminary to warping. In this chapter we examine the performance of some published registration algorithms on the mouse embryo data.

In addition to the overall performance we wish to determine under what circumstances the different methods fail, gauge their accuracy, and determine the effects of preprocessing such as thresholding and edge enhancement. Since fully automatic methods are to be preferred, algorithms which require operator interaction or fiducial markers have been excluded from this study.

Five algorithms, all of which have been briefly described in chapter 3, were chosen[1]: the original algorithm proposed by Hibbard *et al.* [41, 43], that described by Apicella *et al.* [2], that described by Herbin *et al.* [39], and two other methods which form part of the algorithm described by Hibbard *et al.*: centroid and principal axes alignment, and centroid and principal axes alignment followed by cross-correlation to improve the translational alignment. The last two methods were included in order to see how the different parts of the method described by Hibbard *et al.* interact to improve the registration. In particular we wanted to see how crucial the accuracy of the principal axes and centroid alignment is to the success of the registration, and to see if iteration significantly improves the results.

Preliminary results of the method proposed by Herbin *et al.* showed that although this method works well for transformed copies of images (see below for the experimental method), it does not cope well with serial section images. The reason for this is that the DSC criterion essentially counts the number of pixels which have values within a constant (taken to be 5) of their corresponding pixels (of the same coordinates) in the other image. Since, in our images, the background values vary little compared to grey values within the object[2], the background areas of the images match better than the object regions. Therefore, instead of the maximum of the criterion occurring when the two objects maximally overlap, it occurs when the background maximally overlaps. This behaviour is ideal for the application for which the method was designed, that is matching biomedical images of the same lesion taken at different times where the background does not change and a suitable window is (manually) defined and matched to the other image, but it is unsuitable for matching serial section images.

We describe our particular implementations of the remaining four registration algorithms and test them on data from a 9 day mouse embryo. Since the algorithms proposed by Hibbard *et al.* and Apicella *et al.* were discussed in chapter 3 they will not

[1] These five methods were chosen primarily because at the time this work was started, there were no other published automatic registration algorithms. Since it was found that two of these methods produce results of sufficient quality to be passed to the warping algorithm, and because the quality of the results of other registration algorithms seem similar, no other registration algorithms were tested. Nevertheless the experiments performed in this chapter provide insight into the problems of automatic registration, and the merits of other approaches are discussed at the end of the chapter.

[2] This is because the images are captured directly from the microscope and *shade-corrected* which means the background is uniform (apart from noise).

be discussed in detail here.

4.2 Methods

The sections used to test the algorithms were of a 9 day old mouse embryo selected from the section sequences cut by Kaufman for the paper atlas of mouse development [55]. The embryo was embedded in paraffin wax, sectioned at 7 microns, and stained with haematoxylin and eosin. The sections were digitised at a resolution of 2.2 microns using a Leitz Ortholux microscope with a x4 objective, a Pulnix TM86 CCD array video camera and a Primographics Virtuoso frame grabber on a SUN 470. The images corrected for shading and subsampled to half the original resolution. The subsampled images varied in size from 252 to 325 pixels in the x direction and 108 to 203 pixels in the y direction. Some experiments were performed on images subsampled by a factor of 4. These were obtained by subsampling the half sized images. There were 307 sections in the series and these provided sufficient variation to evaluate the various methods. In every section, the tissue was fully contained in the image, that is the foreground was smaller than the whole image and wholly surrounded by background. In these images the foreground values are displayed as dark pixels which correspond to higher pixel values (proportional to density). Four sections from the series are shown as examples in figure 4.1.

The four registration algorithms which were tested were based on principal axes and centroid alignment, or cross-correlation, or both. Polar resampling, for calculating the rotation, was implemented using nearest neighbour values and a resolution of 0.7 degrees (that is 360/512 degrees). A description of the particular implementation of each algorithm is given below, followed by the experiments used for evaluation.

Algorithm 1 (CP)

The first algorithm involved registering the sections by calculating and aligning the principal axes and centres of mass of the images. The principal axes and centres of mass were calculated directly from the greylevels of the images. When thresholded images (see below) were used the algorithm treated them as if they were binary images.

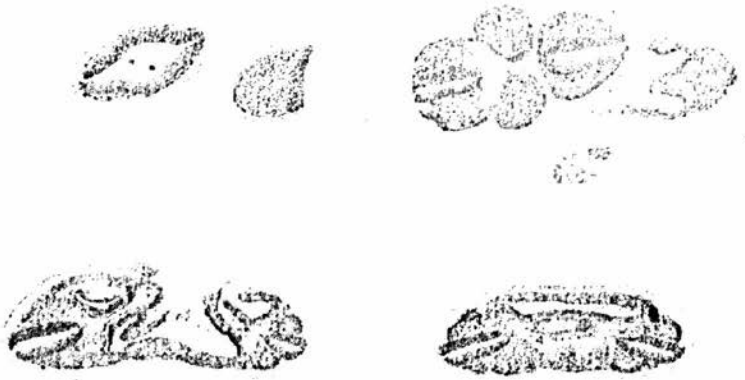


Figure 4.1: Four example sections taken from the 9 day mouse embryo series.

This algorithm on its own is not suitable for registering serial sections because there are two possible alignments of the principal axes. For the application of registering serial sections we will assume the minimum rotation, that is we will assume that the sections do not require a rotation of more than 90 degrees to bring them into register.

Algorithm 2 (XP)

The second algorithm tested is an extension of CP. It consists of principal axes and centre of mass alignment to recover the rotation. This was followed by the use of cross-correlation to improve the translation estimate.

Algorithm 3 (HIB)

An algorithm very similar to the original algorithm described by Hibbard *et al.* [41, 43] was implemented. The differences between the implementation here and that described by Hibbard *et al.* are as follows: In our implementation

- the background pixels were not manually removed from the images;

- the principal axes and centres of mass were calculated directly from the greylevel images and when thresholded images (see below) were considered, they were treated as if they were binary images in line with the method proposed by Hibbard *et al.*;
- to take account of the fact that the principal axes alignment could be 180 degrees out both angles were generated and subsequent processing was performed on both possibilities, the best, in terms of correlation, being selected;
- all rotation was performed using nearest neighbour values; and
- only one resolution was used for polar resampling (0.7 degrees).

In addition, the algorithm stopped either when there was no change after an iteration, or when a maximum number of iterations was reached. The maximum number of iterations was fixed at 5 for transformed copies of identical images (see below) and 10 for serial section images, although in most cases the maximum number of iterations was not reached. Due to digitisation any image rotation algorithm will cause the image to deteriorate on repeated application of the algorithm. If the values of the new image are obtained by interpolation, the resulting image will be blurred and successive applications will increase the blurring. On the other hand if the new pixel values are obtained using nearest neighbour values, successive application of the algorithm may cause pixels to migrate. To prevent deterioration of the image due to repeated rotations, the product of successive transformations (rotations and translations) was calculated and applied to the original version of the transformed image before each iteration. Note that this is not a problem for the other methods because they all involve only a single rotation.

Algorithm 4 (AP)

The final algorithm which was tested, was that described by Apicella *et al* [2], implemented without the scaling component. The only differences are that nearest neighbour values, instead of bilinear interpolation, were used for calculating rotated images and the resolution of polar resampling was finer in this implementation. A drawback of this method is that the images have to have side length a power of two. To achieve this both

images were padded out with randomly generated background values using the mean and standard deviation of the background values of the images. Apicella *et al.* did not have this problem as their images already had side lengths a power of two.

The Hanning window was not implemented as the results for the transformed images (see below) turned out to be adequate. The problem with implementing the Hanning window (or any other window) is that the power spectrum will then depend on the position of the object in relation to the window.

Experimental Method 1 (M1)

The first experimental method involved registering image pairs constructed from known transformations of a single image. For this method, 22 individual quarter size images were selected at regular intervals from the series. New images were generated by either rotating or translating these images (see below). The series of rotations varied from -180 degrees to 180 degrees with intervals of 10 degrees, except for CP where the rotations varied from -90 to 90 degrees; the series of translations varied from -50 pixels to +50 pixels in both the x and y directions. Both the x and y intervals were of 5 pixels and an increment in either the x or y direction was performed for each new test translation.

Experimental Method 2 (M2)

The second experimental method involved registering image sequences from real data. Some initial transformations were performed on the image sequences before applying this experimental method because when an object is rotated and translated, the magnitude of the translation component depends on the centre of rotation. If an object is rotated about its centre then the translation will represent the distance from the new position of the centre of the object to its new position. But if an object is rotated about a point other than its centre the position of the object will change and the translation will clearly not represent the distance that the object has moved from its original position. Therefore, to ensure that the results concerned with translation did not depend too heavily on the rotation, each section in the unregistered stack was translated so that the x and y coordinates of the centre of the stack, calculated by considering the stack

as a rectangular block, coincided with the z-axis. Not all 307 sections were included in the experiments since the top and bottom few sections contained very little greylevel information. For this reason some of the top and bottom sections were discarded and the experiments were performed on the remaining 289 sections.

Evaluation of the Results

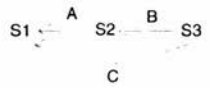
The results of M1 can be evaluated against an absolute truth standard; the results of M2 cannot, but we can compare them with the results of manual registration. To evaluate the performance of the different registration algorithms, the following parameters were considered:

1. the calculated rotation between sections;
2. the calculated translation between sections;
3. the calculated cross-correlation between sections;
4. the continuity of features in planes orthogonal to the original sections: sections orthogonal to the original cutting direction were cut through the stacks of registered sections. Visual evaluation of the resulting sections is subjective but useful given the subsequent application.

Note that sections cut at the same coordinates through stacks of sections registered in different ways may be different due to systematic error in the registration procedures. The calculated rotation and translation were not available for the manually registered sections, but it is still useful to compare these results for automatic registration to see whether the algorithms produce similar or widely differing transformations.

The above criteria give little indication of the error in rotation and translation of the various methods for real image sequences. It is possible to obtain error estimates for each method by calculating the transformations required to register three consecutive sections. To do this we assume that the errors in registering adjacent sections are similar to those for registering two sections which are separated by a single section. Given sections S1, S2 and S3, let A be the calculated transformation which will transform S2

so that S1 and S2 are registered. Let B be the calculated transformation which will transform S3 so that S2 and S3 are registered and let C be the calculated transformation which will transform S3 so that S1 and S3 are registered.



Then if there are no errors in the transformations,

$$BAC^{-1} = I,$$

where I is the identity transformation. However, there will be errors in the transformations so

$$BAC^{-1} = E,$$

where E is the error transformation. If we assume that the mean and standard deviation of the errors in each of the transformations are the same, then it can be shown that the mean of the rotation in E is the same as the means of the errors in the rotations of A, B, and C, and that the standard deviation of the error in the rotation in E is $\sqrt{3}$ times the standard deviation of the error in the rotations of A, B and C. Therefore, given the mean and standard deviation for E, the mean and standard deviation for the errors in the rotation can be estimated. A similar reasoning does not hold for translation because the translation error in E is not related to the translation errors in A, B and C in a simple way: it depends on the rotations and the errors in rotation. Consequently there is no simple formula which will relate the translation in the error transformation to the error in translation of the individual transformations. Despite this, the mean and standard deviation of the error in translation of the error transformation will provide a useful measure with which to compare the performance of the various registration algorithms.

Preprocessing

All the algorithms were also tested on images which had been preprocessed in various ways to see if this had any effect on the results. The various types of images used in the experiments are given below:

P1 Raw, greylevel images.

P2 Background cut images using an automatically generated threshold. The values above threshold were left unchanged[3]. The threshold was generated by analysis of the histogram of the grey values. The slope of the histogram was calculated at each point above the histogram peak (assumed to represent background values) and the threshold was set at the first point where the slope becomes zero or positive.

P3 Edge enhanced images. Images were enhanced using the Roberts cross operator [33], the Sobel operator [33] or a standard deviation filter [103].

P4 The original image was rotated or translated and the result was embedded in a rectangular image large enough to contain both the original and rotated images.

P5 The original image was automatically thresholded as above, then a new image was made by rotating the thresholded image.

P4 and P5 are designed for use with M1. The new images obtained using P4 were padded out with random background values obtained from the mean and standard deviation of the background grey values. The mean and standard deviation for the greylevels was determined interactively and the same values were used for all images. This process was used to make a pair of M1 images simulate a pair of serial section images. When large translations were performed this resulted in large images which consisted mostly of background and therefore gave good test images on which to gauge the effect that large numbers of background values have on the various algorithms.

The algorithms were tested on edge enhanced images because Hibbard *et al.* had reported improved registration in this case. The standard deviation filter was included in the preprocessing algorithms because it emphasises the regions of change in the image. In each case, a 7×7 neighbourhood was employed because it smooths the image and picks out the larger features which are more likely to persist between images.

4.2.1 Registration Experiments

[3] Note that in all the images considered here the grey values are higher for dark (foreground) regions and low for light (background) regions.

	(M1, P4)	(M1, P5)	(M2, P1)	(M2, P2)	(M2, P3)
CP	CP1.4	CP1.5	CP2.1	CP2.2	—
XP	—	—	XP2.1	XP2.2	XP2.3
HIB	HIB1.4	HIB1.5	HIB2.1	HIB2.2	HIB2.3
AP	AP1.4	—	AP2.1	—	AP2.3

Table 4.1: Table of experiments.

The four algorithms were tested on variously preprocessed images. The actual experiments performed are listed in table 4.1. The experimental methods and image preprocessing methods are given across the top of the table, and the registration algorithms are listed in the left hand column. Each experiment was assigned a name depending on the algorithm, the experimental method and the image preprocessing method used.

CP was not applied to edge enhanced images because this would be inappropriate as it is designed for solid objects rather than edges. AP was not applied to thresholded images because it was specifically designed for greylevel images. XP was not tested on transformed images as the results for CP showed that if the correct rotation was found, the correct translation was also found and therefore no further translation correction was required.

4.3 Results

We now give the results of applying the four algorithms to the various types of images. The results are presented as tables, graphs and images of sections cut through the registered stack of sections. The results for positive rotations only are shown in the graphs as negative rotations gave similar results. The graphs which show the error in the calculated rotation, give only the modulus of the error. In all cases the correlations shown are $(1 - correlation)$. Note that, in most cases, the results for the series of sections are shown for the middle 200 sections only as the methods failed for some sections at the begining and end of the series.

4.3.1 Principal Axes

Transformed Images

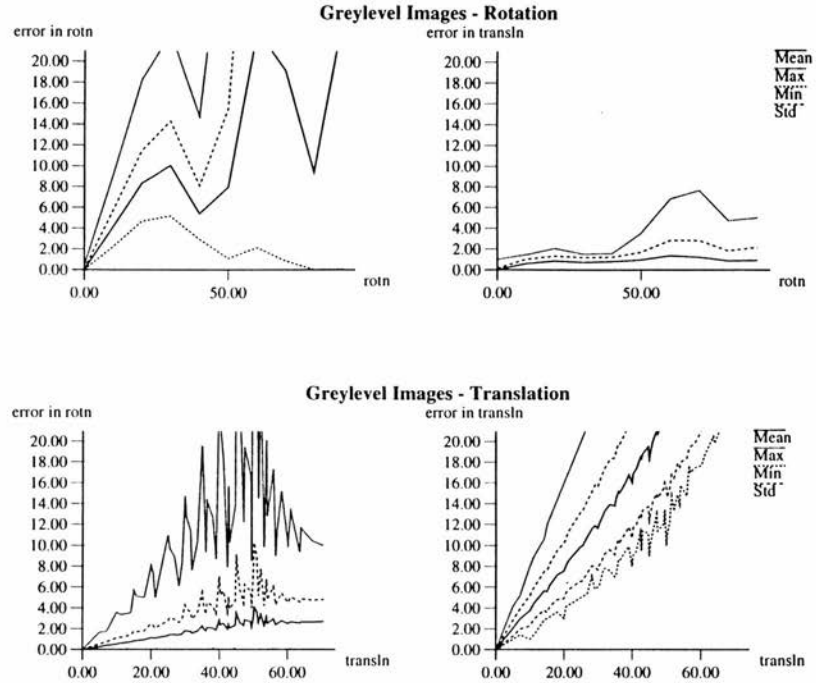


Figure 4.2: Graphs showing the errors in the calculated rotation (rotn) and translation (transln) produced by the principal axes alignment tested on rotated (top graphs) and translated (bottom graphs) greylevel images. The rotation and errors in rotation are shown in degrees; the translation and errors in translation are shown in pixels.

The results of applying principal axes and centroid alignment to rotated or translated greylevel images (experiment CP1.4) are shown in figure 4.2. When rotation is applied we see that the error in rotation rapidly becomes large (maximum of 20 degrees or more) but that the error in translation remains fairly small (maximum of 3 pixels up to a rotation of about 50 degrees). When a translation is applied to the greylevel images we see that the error in translation increases rapidly with the applied translation and that the error in rotation is generally within 10 degrees but that it can be larger than 20 degrees.

When the images are automatically thresholded and rotations are applied to the thresholded images (experiment CP1.5), the results are much better. By referring to figure 4.3,

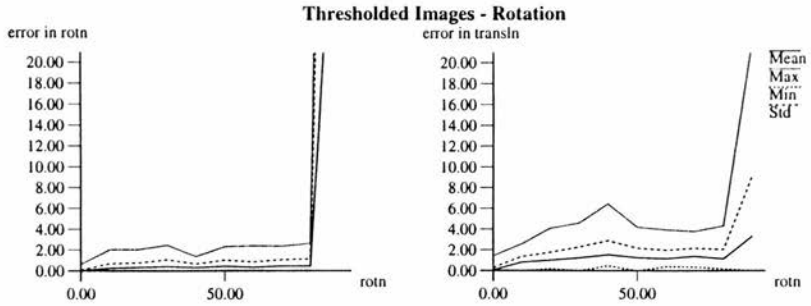


Figure 4.3: Graphs showing the errors in the calculated rotation (rotn) and translation (transln) produced by the principal axes alignment on rotated thresholded images. The rotation and errors in rotation are shown in degrees; the translation and errors in translation are shown in pixels.

	Error in Rotation			Error in Translation		
	mean	std	max	mean	std	max
CP1.5	0.268	0.486	2.636	0.884	1.034	7.401

Table 4.2: Table of results of applying CP to rotated, thresholded images. The error in rotation is given in degrees; the error in translation is given in pixels.

we see that the errors in rotation are small up to about 60 or 80 degrees. The mean and standard deviation of the errors obtained for angles between -80 degrees and 80 degrees are given in table 4.2. From this table we see that the errors in rotation are small (up to a maximum of about 2.6 degrees) and that the error in translation is generally small (about 2 pixels) but that it can be as large as 7.4 pixels. The results for translated thresholded images were perfect in that no errors were found in the calculated translation.

From these results we can conclude that most of the errors obtained for the greylevel images are due to the effect of the background values in these images. As the translation increases, the size of the image increases and thus the area containing background values increases. This has the effect of pulling the centre of mass towards the centre of the image and of pulling the principal axes towards the principal axes of a rectangle, though from figure 4.2 we see that the centre of mass is more affected than the directions of the principal axes.

In order to see how the background affects the angle of the principal axes, a further experiment was performed. The principal axes were calculated for both the original and thresholded version of all the images in the 9 day mouse embryo series. Then the modulus of the difference in the angles of the main principal axes was calculated and plotted against the percentage of object pixels (those higher than the threshold) to the number of pixels in the greylevel image. It was found that if the percentage of object pixels was greater than about 30% the difference between the angles was within 1 degree; if the object made up 20% of the image, the difference was within 6 degrees; and if the object made up only 10% of the image the difference was within 10 degrees. When the object made up less than 10% the direction of the main principal axis was close to that of the long axis of the image. From this we may conclude that principal axes alignment is unreliable unless at least 30% of each image consists of object pixels. However adjacent serial sections are likely to consist of similar objects with similar numbers of background pixels and therefore, if the object makes up more than 10% of the image, the angles of the principal axes will be affected in similar ways causing systematic rather than random registration errors. Thus the visual quality of the registration may not be affected. Note that this analysis could not be meaningfully applied to the centre of mass because we can expect the centre of the object to be close to the centre of the image.

Serial Section Images

	1 - Correlation		
	mean	std	max
CP2.1	0.220	0.068	0.484
CP2.2	0.314	0.109	0.600

Table 4.3: Correlation values for the results of applying CP to greylevel (CP2.1) and thresholded (CP2.2) serial section images.

The results of applying CP to greylevel (CP2.1) and thresholded (CP2.2) serial section images are given in table 4.3 and figure 4.4. The results given in the table are for the series of 200 sections taken from the middle of the series. The results for the whole series were not so good because the registration method had trouble with the sections at the ends. In particular the method failed in the case where one section contained

No Registration

Principal Axes

Figure 4.4: The left image shows a section of the unregistered stack; the right image shows the section with the same coordinates of the stack registered using centre of mass and principal axes alignment (on the original greylevel images).

one object and the next section contained two similar objects. In addition it tended to fail when the object made up less than 10% of the image. If we examine figure 4.4, which shows perpendicular sections through the unregistered and registered (using the original greylevel images) stack, we see that although the registration procedure has improved on the section obtained before registration, the result is still not very good in that the structure is not very clear. A cut through the stack of sections produced using thresholded images is not shown but the results were no better. From table 4.3 we see that the correlation values obtained for the thresholded images are worse than those obtained for the original greylevel images suggesting that the reconstruction is of a poorer quality. The reason for this may be due to variations in the areas of the thresholded images caused by the automatic thresholding algorithm.

4.3.2 Principal Axes and Cross-Correlation

Serial Section Images

XP was applied to real image sequences only. The results of applying this algorithm to greylevel (experiment XP2.1), thresholded (experiment XP2.2) and edge enhanced (experiment XP2.3) images are given in table 4.4, and in figure 4.5. The results given in the table are the statistics calculated for the series of 200 sections selected from the middle of the whole series.

Comparing the errors found in the translation calculation given in table 4.4 we see that

	1 - Correlation			Error in Translation		
	mean	std	max	mean	std	max
XP2.1	0.165	0.050	0.294	0.685	0.772	4.972
XP2.2	0.307	0.112	0.630	1.056	1.091	8.596
XP2.3 (rob)	0.258	0.090	0.563	1.024	1.295	8.976
XP2.3 (sob)	0.247	0.091	0.587	0.847	0.899	5.563
XP2.3 (std)	0.255	0.095	0.585	0.475	0.658	4.129

Table 4.4: Error measurements for XP applied to greylevel (XP2.1), thresholded (XP2.2) and edge enhanced (XP2.3) serial section images. The correlation values for experiments XP2.2 and XP2.3 are those found for unprocessed images after the registration transformations have been applied. The error in translation is given in pixels.

the lowest values are obtained for the images preprocessed with the standard deviation operator (XP2.3 (std)) but that the values obtained for the greylevel images (XP2.1) are also good. Note that there were no errors in the calculated rotation because principal axes alignment cannot give different transformations when calculated directly and when calculated from a series of transformations.

The correlation values in table 4.4 suggest that registration using the original greylevel images gives the best results and that registration using thresholded images gives the worst results. When the sections through the registered stack are compared (figure 4.5), we find that the results produced for thresholded images are significantly worse than the others. There is not a lot to choose between the results of the other experiments, although the results for the greylevel images may be slightly better. All show some regions where the registration seems to have worked well and regions where the structure is not very clear. The results for images filtered with the Sobel operator are not shown because they were very close to those obtained using the Robert's cross filter.

4.3.3 Hibbard's method

Transformed Images

The results of applying HIB to transformed greylevel images (experiment HIB1.4) are given in figure 4.6. From figure 4.6 we see that this algorithm breaks down for applied rotations which are greater than 50 degrees. The mean and standard deviation for

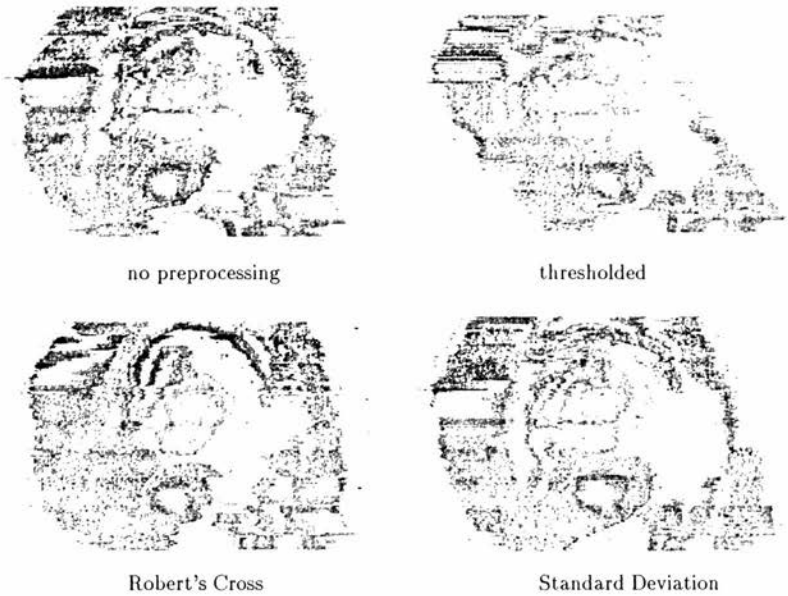


Figure 4.5: Results of registering serial sections using the principal axes and cross-correlation algorithm.

	Error in Rotation			Error in Translation		
	mean	std	max	mean	std	max
HIB1.4	0.281	1.560	39.037	0.352	0.842	13.416
HIB1.5	0.240	0.247	1.486	0.757	0.380	2.235

Table 4.5: The results of applying HIB to both greylevel (HIB1.4) and thresholded (HIB1.5) images which have been rotated. The values for the greylevel images are the values obtained for rotations between -50 degrees and 50 degrees inclusive. The error in rotation is given in degrees; the error in translation is given in pixels.

the errors in rotation and translation for applied rotations between -50 degrees and 50 degrees are given in table 4.5. We see that these errors are generally less than 2 degrees and 2 pixels. The maximum values of 40 degrees and 14 pixels arise from a single poorly registered image. Examining figure 4.6 we see that this method also breaks down for applied translations which are greater than 30 pixels. The mean and standard deviation for the errors in rotation and translation for applied translations of up to and including 29 pixels are given in table 4.6. Again, the errors are generally small and large errors

	Error in Rotation			Error in Translation		
	mean	std	max	mean	std	max
HIB1.4	0.111	0.344	10.925	0.187	0.390	4.489

Table 4.6: The results of applying HIB to translated greylevel images. The values given are those obtained for translations between -29 pixels and 29 pixels inclusive. The error in rotation is shown in degrees; the error in translation is shown in pixels.

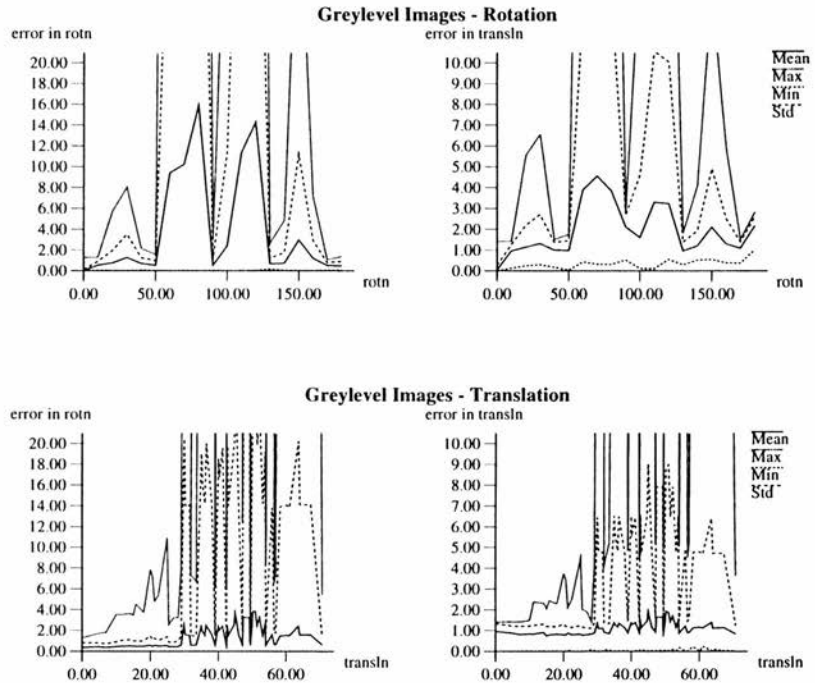


Figure 4.6: Graphs showing the error in the calculated rotation (rotn) and translation (transln) for HIB applied to rotated (top graphs) and translated (bottom graphs) greylevel images. The rotation and errors in rotation are shown in degrees; the translation and errors in translation are shown in pixels.

	1 - Correlation			Error in Rotation			Error in Translation		
	mean	std	max	mean	std	max	mean	std	max
HIB2.1	0.150	0.043	0.271	0.283	0.399	2.109	0.677	0.508	2.323
HIB2.2	0.270	0.111	0.607	0.290	0.696	6.328	0.846	0.702	6.649
HIB2.3 (rob)	0.264	0.103	0.621	0.515	0.590	2.813	1.048	0.858	5.227
HIB2.3 (sob)	0.261	0.105	0.690	0.444	0.540	3.516	0.900	0.753	4.616
HIB2.3 (std)	0.274	0.104	0.599	0.245	0.365	2.109	0.588	0.464	3.010

Table 4.7: Error measurements for the results of applying HIB to greylevel (HIB2.1), thresholded (HIB2.2), and edge enhanced (HIB2.3) serial section images. The correlation values for experiments HIB2.2 and HIB2.3 are those found for unprocessed images after the registration transformations have been applied. The errors in rotation and translation are shown in degrees and pixels respectively.

in rotation coincide with large errors in translation.

The mean and standard deviation for the results obtained from rotated thresholded images (experiment HIB1.5) are also given in table 4.5. From this, we see that all errors are small for rotated thresholded images. Note that there were no errors for translated thresholded images. We can therefore conclude that the large errors obtained for greylevel images for large rotations and translations are due to the effects of the background values.

Serial Section Images

The results of applying HIB to greylevel (HIB2.1), thresholded (HIB2.2), and edge enhanced (HIB2.3) serial section images are given in figure 4.7 and in table 4.7. The values given in the table are for the series of 200 sections from the middle of the whole series.

Examining table 4.7 which gives the mean, standard deviation and maximum values for the various error measurements (correlation, rotation and translation) we see that the errors in the calculated rotations and translations are generally small although they can be as large as a few degrees or a few pixels. When the correlation values are examined, we see that greylevel images (HIB2.1) give the best correlation values. However on examination of the sections cut through the registered stacks shown in figure 4.7, the section through the stack produced using images filtered with the standard deviation

no registration

no preprocessing

Robert's Cross

Standard Deviation

Figure 4.7: The results of registering the stack using Hibbard's method on variously preprocessed images. The results for images filtered with the Sobel operator are not shown because they were very close to those produced with images filtered with the Robert's cross operator.

	mean	std	max
HIB2.1	1.96	1.29	8
HIB2.2	1.86	1.76	10
HIB2.3 (rob)	1.82	1.13	7
HIB2.3 (sob)	1.91	1.06	5
HIB2.3 (std)	1.92	1.18	10

Table 4.8: Number of iterations using HIB on serial section images.

operator (HIB2.3 (std)) seems to be slightly better than the others. The structure on the right hand side is joined up and there is more well defined structure in the centre of the section. This shows that correlation may not be a very accurate indication of the quality of a reconstruction.

This method was run with a constraint of a maximum of 10 iterations. The maximum number of iterations was reached for one section only when using images preprocessed with the standard deviation operator, and for two sections when using thresholded images. The results for all the images are given in table 4.8.

4.3.4 Apicella's Method

Transformed Images

	Error in Rotation			Error in Translation		
	mean	std	max	mean	std	max
AP4.1	0.333	0.315	1.641	0.771	0.510	1.550

Table 4.9: Table showing the errors for AP applied to rotated greylevel images. The error in rotation is given in degrees; the error in translation is given in pixels.

The results of applying AP to rotated images (experiment AP1.4) are given in table 4.9; there were no errors when the images were translated. From table 4.9 we see that the errors in rotation were within 2 degrees and that the errors in translation were within 2 pixels.

Serial Section Images

The results of applying AP to greylevel (AP2.1) and edge enhanced (AP2.3) serial section images are given in figure 4.8 and in table 4.10. The values in the table are calculated for the series of 200 sections from the middle of the whole series to avoid the effects of small amounts of image data. In particular the method failed in the case where adjacent sections contained different numbers of objects.

Examining the table we see that the images preprocessed with the standard deviation operator give the lowest errors in the estimated rotation but that the images with no

	1 - Correlation			Error in Rotation			Error in Translation		
	mean	std	max	mean	std	max	mean	std	max
AP2.1	0.162	0.049	0.278	0.661	0.971	7.031	1.118	1.252	9.862
AP2.3 (rob)	0.208	0.099	0.625	0.728	1.037	4.922	1.675	1.476	7.400
AP2.3 (sob)	0.203	0.098	0.610	0.519	0.977	6.328	1.419	1.492	9.000
AP2.3 (std)	0.188	0.083	0.629	0.205	0.906	5.625	1.070	1.744	12.000

Table 4.10: Error measurements for AP applied to greylevel (AP2.1) and edge enhanced (AP2.3) serial section images. The correlation values for experiment AP2.3 are those found for unprocessed images after the registration transformations have been applied. The errors in rotation and translation are given in degrees and pixels respectively.

No registration

No preprocessing

Robert's Cross

Standard Deviation

Figure 4.8: The results of registering the stack using Apicella's Method on variously preprocessed images. The results for images filtered with the Sobel operator are not shown because they were very close to those produced with images filtered with the Robert's cross operator.

preprocessing give the lowest errors in the estimated translation. We also see that the best correlation values are obtained for greylevel images and when the sections in figure 4.8 are examined we see that these images clearly produce the best results.

4.3.5 Comparison of the Methods

From the above results we see that XP, HIB and AP all produce reasonable registrations. We may discard CP as it produces results which are clearly inferior. We now compare XP, HIB and AP in more detail together with the results of manually registering the sections.

There are several criteria we can use to compare the various methods. We will consider

- the calculated correction transformation between sections,
- the calculated correlation values,
- the estimated errors in the transformation calculation, and
- subjective assessment of orthogonal sections.

The main problem with comparing the different registration methods is that due to the lack of fiducials or episcopic images and the presence of distortion, the correct result is not known and cannot be known for certain.

Graphs comparing the calculated transformations for the automatic registration methods are shown in figure 4.9. Note that both graphs show the values for every fifth section and the graph showing the calculated translation shows the translation in the x direction. When these graphs are examined we see that the various automatic registration algorithms produce similar values for translation but that there is some disagreement in the values of the rotations.

The mean, standard deviation and maximum for the errors in the transformation are given in table 4.11. From this table we see that the method which produces the lowest error in rotation is XP which finds no errors because principal axes alignment was used to calculate the rotation. HIB (with images filtered with the standard deviation operator) has errors which are significantly lower than those found for AP. When the

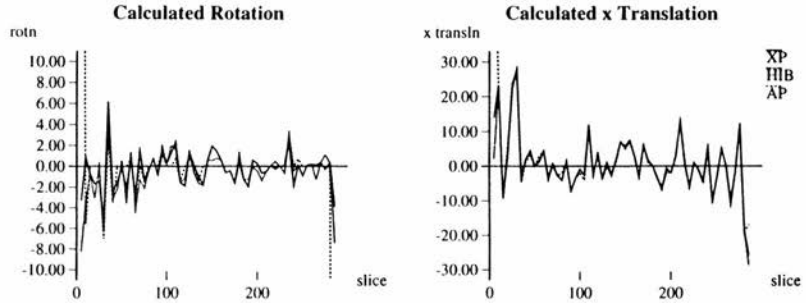


Figure 4.9: Graphs showing the calculated rotation and translation for every 5th section calculated by XP, IIIB and AP. The graph showing the calculated translation shows the x translation only. The rotation and translation are given in degrees and pixels respectively.

	1 - Correlation			Error in Rotation			Error in Translation		
	mean	std	max	mean	std	max	mean	std	max
CP	0.220	0.068	0.484	0	0	0	0	0	0
XP	0.165	0.050	0.294	0	0	0	0.685	0.772	4.972
IIIB (std)	0.274	0.104	0.599	0.245	0.365	2.109	0.588	0.464	3.010
AP	0.162	0.049	0.278	0.661	0.971	7.031	1.118	1.252	9.862

Table 4.11: Table of error measurements for the four registration algorithms. The correlation values are those found for unprocessed images after registration transformations have been applied. The errors in rotation and translation are given in degrees and pixels respectively.

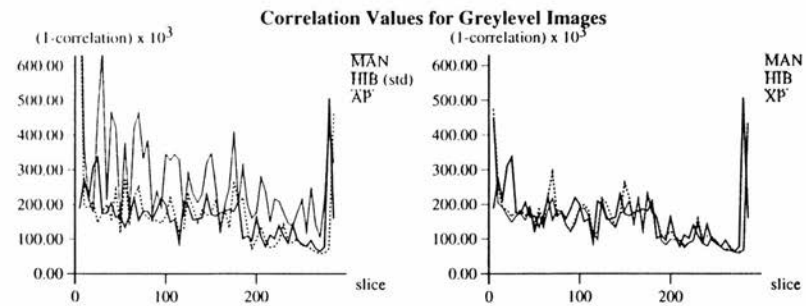


Figure 4.10: Graphs showing the correlation values for each section calculated from the registered stacks of greylevel images.

Manual

Hibbard (std)

Apicella

Principal Axes and Cross-Correlation

Figure 4.11: Sections perpendicular to the x-axis.

error in translation is examined we see that (of XP, HIB and AP) HIB with images filtered with the standard deviation method produces the least errors.

Graphs of the correlation values are given in figure 4.10. The values are those obtained by calculating the correlation values between unprocessed images registered using the appropriate registration transformations. The left hand graph compares the correlation values between the various registration algorithms where Hibbard's method was performed on images which had been filtered with the standard deviation operator; the right hand graph shows the values for Hibard's method with the original greylevel images. By comparing these two graphs we see that the correlation values for the greylevel

Manual

Hibbard (std)

Apicella

Principal Axes and Cross-Correlation

Figure 4.12: Sections perpendicular to the y-axis.

images are similar for the manual method and the automatic registration algorithms, but that although Hibbard's method filtered with the standard deviation operator seemed to give the best overall result, the correlation values are noticeably worse.

Cross sections through the registered stacks of images are given in figures 4.11 and 4.12. Figure 4.11 shows sections perpendicular to the x-axis; figure 4.12 shows sections perpendicular to the y-axis. In all cases except where the sections were registered manually the sections were made with the same coordinates. When these images are examined we see that the different automatic registration images all produce slightly different cross sections in that there are differences in the structure shown in the images. On close examination of the images we can conclude that Hibbard's method and manual registration produce the best results from a visual point of view. In these images more structure is clearly defined. The reason for the comparatively poor results using Apicella's method may be because the rotation is calculated by a correlation of a 1D array of values obtained by summing along radial axes. The method of correlating 2D arrays seems to give more accurate results. In all the images the boundaries of

the structures are jagged. This may be due to deformation in the sections, which would make it impossible to produce a smooth reconstruction using only rotation and translation.

The automatic registration algorithms did not produce good results for all the sections. In particular, all the methods produced poor results when given one section with one object followed by a section with two similar objects. Also, the principal axes alignment will not be able to cope with circular sections, which may cause Hibbard's method to fail. However clearly, all automatic algorithms will have trouble with symmetric sections.

The different registration algorithms run at different speeds depending on the number of steps in the registration process. Hibbard's method is the slowest; the principal axes and cross-correlation method is the fastest. For the sections on which the algorithms were tested, the principal axes and cross-correlation method is the best choice because it is the fastest and produces similar results to the other two methods. If speed is less important Hibbard's method can be chosen for its extra quality in alignment.

4.4 Discussion

In the previous section we concluded that of all the automatic registration algorithms tested, Hibbard's method (HIB) gives the best results in terms of visual criteria. However the results of the method which uses principal axes aligment and cross correlation to correct the translation (XP) were very similar. The fact that XP and HIB give similar results shows that iteration does not improve the results of registering these images very much. We can also conclude that the rotation computed by the principal axes alignment is accurate, but the centre of mass alignment does not give accurate results. This agrees with the results of experiment CP1.4 where it was found that background values affect the position of the centre of mass more than the directions of the principal axes.

From table 4.11 we see that principal axes alignment (CP) gives better correlation values than HIB although the visual results are much worse. Also, although the visual results given by XP are similar to those obtained by HIB, the correlation results are significantly better. The above suggests that cross-correlation is not a good measure of

the quality of the registration.

Cross-correlation scores grey values in corresponding positions so that higher values are obtained if the grey values are identical. Unfortunately since consecutive serial sections are, in general, different we cannot expect the features to match up in this way. For example it is unlikely that the positions of prominent edges will correspond exactly. In the case where the sections consist of concentric rings, for example, where the radii are sufficiently different that the edge regions do not overlap, cross-correlation will not give the correct result. To produce the correct result in this case we would need to match individual points in the images and then calculate a transformation which minimises the distances between corresponding points. This is, in effect, the algorithm proposed by Toga and Banerjee [99] who match grid points to areas in the adjacent image using cross-correlation. However Toga and Banerjee found that this approach did not significantly improve the registration results over cross-correlation. The reason for this may be because, as we shall see later, cross-correlation is not a good method for finding correspondences as the best match will tend to move along an edge. In addition by matching regularly spaced grid points, some points, such as edge points, for which the best match is more likely to give the correct correspondence, may not be chosen, and points in homogeneous regions which provide little alignment information may dominate. Better results would be obtained by choosing significant points, such as edge points, to match. However before drawing any conclusions, this should be tested, preferably with a better algorithm for finding the correspondences such as that developed in chapters 7 and 8.

Alternative match criteria were proposed by Rydmark *et al.* [85] and Andreason *et al.* [1]. The criterion proposed by Andreason *et al.* suffers from the same problems as cross-correlation since the difference between pixels in corresponding positions are taken. Rydmark *et al.* use a criterion which is essentially an overlap measure, and as such will also not cope with sections consisting of concentric rings.

The automatic registration algorithms tested above did not produce good results for all the sections. In particular all the methods failed when given one section with one object followed by a section with two similar objects. They also failed when confronted with two consecutive sections containing objects of different sizes and shape. With

the possible exception of the method proposed by Toga and Banerjee, these failures are common to all the registration algorithms proposed to date. The iterative part of Hibbard's algorithm can cope with this problem as long as the two sections are roughly aligned first. In a later version of their algorithm, Hibbard *et al.* [40, 42] have attempted to address this problem. To pre-empt failure from the principal axes alignment, the rotational component is initially calculated from the autocorrelations of the images. This is very similar to calculating the rotation from the Fourier spectrum as proposed by Apicella *et al.* and suggests that the results of Appicella's method could be improved by subsequent iteration. In addition the images are smoothed using a low pass filter to remove the finer details. This was found to reduce the number of failures, but user interaction was still necessary.

Because of the problems described above, automatic registration will have to be checked by the operator. Generally when the sections match well the correlation value is relatively high, whereas when the registration has failed the correlation value is low. Therefore the correlation value could give a human operator an indication as to the quality of the registration for each section.

None of the methods tested above will cope well with sections where the tissue is not fully contained in the image. The principal axes alignment will clearly fail if the field of view is filled with tissue; the other methods will fail because the Fourier transform will be convolved with the square (or rectangular) image. However windowing as suggested by Apicella *et al.* may solve this problem as it gives more weight to the features positioned in the centre of the images and less weight to those near the edge. It is safe to assume that the object of interest will be located in the centre of the image. The problem with windowing, however, is that by giving more weight to pixels close to the centre of the image, leverage from good matches close to the edges of the images is reduced leading to possible errors in the calculated rotation. In view of these problems, the algorithms proposed by Rydmark *et al.* and Andreason *et al.* should possibly be considered for sections of this type, as these methods are suited to images where the object fills the field of view.

We note that shifts and twists in the final reconstruction with respect to the original object are likely and are unavoidable. This is due to the information lost by sectioning

as discussed in chapter 2. Due to registration errors, cumulative registration error is also possible. This problem of cumulative error was addressed by Andreason *et al.* who weight each transformation by previous transformations, producing significant improvements in the registration. However the only reliable way of reconstructing the original object is to include spatially mapped fiducials or refer to episcopic images as practised by Lamers and Laan *et al.* [60, 59] and Verbeek [100].

All of the published registration methods appear to produce reconstructions of similar quality. The structure is clearly visible, but the surfaces of the structures are jagged. One solution is to remove data [35, 24]. However a better approach to improve the quality of the reconstruction is to correct local deformations. A method for correcting these deformations will be discussed in the next chapters. However we shall see that although the algorithm can produce a reasonable reconstruction from unregistered images, the most useful results are obtained from registered images. In this chapter we have shown that published registration algorithms can be successfully applied to our sections.

Chapter 5

Introduction to Warping.

In chapter 4 we established the need for warping. In this chapter we will consider the warping problem and give an overview of the way this problem has been tackled and an indication why the problem has been approached in this particular way. We will also discuss some of the problems which must be solved to achieve our objectives.

5.1 Requirements

First we consider the requirements of warping. The aim is not to warp the object so that it faithfully matches the original object before sectioning, but to correct the relative deformation of the sections so that the various 3D surfaces are smooth. Once the image has been warped, the conformance to the original object can be achieved interactively by allowing the user to apply external constraints. Indeed this would allow the object to be straightened out if it has been bent by the histological preparation process. It is the intention that the warping process should be fully automatic and that the techniques developed should be applicable to a wide range of serial section reconstruction problems. However in this study the specific problems caused by tears and folds will not be considered. Note that, for warping, we assume that the sections are registered.

There have been few recent studies on the deformation undergone by the sections as they are processed, but it is generally suspected that although good results have been obtained by global linear warps [59, 49, 104], the deformations are local in nature

[10, 51]. In his warping technique, Dürr [26] assumes that the deformations are local. The work done by Laan *et al* [59] which employed a global linear warp was extended by Verbeek [100] to warp contours locally using local information. Olivo *et al* [72] found that the deformation could not always be adequately corrected by a global polynomial warp function but found that a subsequent local correction gave good results. The study published by Jones *et al.* [51] on the distortions undergone by the sections, found that the distortion of wax sections is tissue specific and that the distortions of individual sections within a series vary significantly. In contrast, in their study of plastic sections, Berthold *et al.* [10] concluded that the tissue was uniformly compressed. These studies show that the deformation undergone by the sections depends on a variety of factors including the type of tissue and embedding material. Therefore, and because a local warp function will also describe all other kinds of warps, we require that the warp function be calculated locally for each part of the section. Nevertheless we will analyse the results of the warping process to see what kinds of deformation have been corrected.

It is important that the warp function defined over each section should be continuous because any discontinuity would correspond to a tear in the tissue. We require that large warps are associated with a large cost, and that large warps are constrained so that displacements which would produce tears in the tissue are avoided. An effective way of satisfying these criteria is to model the sections as thin elastic plates. The elastic properties will ensure that the warp is both continuous and that large warps are associated with a large cost, corresponding to large forces acting on the section. Also different parts of the object can be made to behave differently by assigning different material properties to those parts.

In order to perform a warp based on local warp functions we require local information. This implies that it is necessary to find a lot of corresponding points. In fact the number of corresponding points required make interactive warping impractical because of the difficulties in automatically establishing enough exact correspondences for each section. Finding full correspondence is an unsolved problem for general biological images although there are efficient algorithms for some distinct situations such as for the case where the sections consist of individual cells [105, 106], or in some cases external boundaries can be automatically extracted [100]. However, since we prefer that the

methods developed for this project be applicable to a wide range of problems, we would prefer not to match one set of segmented features to another. Instead we match feature attributes from features extracted from one image directly to another, possibly filtered, image.

Most of the warping techniques developed in the literature are designed to bring correspondence points into exact alignment so that one image is warped so that it matches another image exactly [26, 100, 14]. However in order to produce a smooth progression from the first section to the last section it is important that the warp calculation for each section obtains information from both adjacent sections, and that this information is updated as these sections are warped. In this way the position of a point on one section should finish up between the positions of its corresponding points on adjacent sections. Since we intend to model the sections as thin elastic plates, forces have to be applied to the sections. A simple way of doing this is to link the corresponding points with springs and allow the springs to pull the corresponding points towards each other. This has the advantage that the forces exerted by the springs change as the sections are warped. Note that the springs are assumed to have zero length under zero tension.

In summary, the solution to the warping problem examined in this thesis is to model each section as a thin elastic plate constrained so that points in the plate can move only in the x and y directions. Then we attach springs to corresponding points in adjacent sections and let go and wait until the whole stack reaches an equilibrium. There are several problems to be overcome with this approach. First we have to determine a mathematical model for modelling the sections as thin elastic plates so that each section can be warped in an appropriate way, then we have to determine a way of propagating the warp for each section and calculating an equilibrium for the whole stack. The most difficult problem is that of calculating the correspondences between adjacent sections. We will discuss these problems in more detail below and in subsequent chapters.

5.2 Problems and Issues

5.2.1 Mathematical Modelling

The sections will be modelled as thin, perfectly elastic plates. They will be perfectly elastic in that any warp will be totally recoverable and the sections will not tear as a result of the warping. To warp each plate we are given the forces, exerted by the springs, acting on the plate and we wish to calculate the displacements of all the points in the plate. There are two sources of energy in this application: the work done by the springs and the potential energy of the stresses and strains within the plate. We need the displacements when the section is in equilibrium, that is when the forces in the plate and the springs are equal and opposite. The total energy of the system, $\mathcal{E}$, is the sum of the internal energy in the plate and the internal energy in the springs, which can be written as [81, 82]

$$\mathcal{E} = \frac{1}{2}\int_V \boldsymbol{\sigma}^T \boldsymbol{\epsilon} dV + \frac{1}{2}\sum l_i^2 k_i,$$

where $\boldsymbol{\sigma}$ is the stress vector, $\boldsymbol{\epsilon}$ is the strain vector, k_i is the spring constant of spring i, l_i is the extension of spring i, and the integration is over the volume of the current configuration of the section. We will refer to this equation as the energy functional.

The energy functional is a non-linear equation because the current configuration of the section is unknown, and thus the volume over which we are integrating depends on the displacements which we are trying to calculate. The strain can be calculated from the displacements, but the stress is related to the strain by constitutive relations which need not be linear, introducing another source of non linearity into the equation. The method which is chosen to solve the above equation must be able to cope with these non-linearities.

There are two main ways of solving this equation numerically: the method of finite differences and the finite element method. To use finite differences the energy functional is converted into a partial differential equation (using Euler-Lagrange theory). This partial differential equation can be solved numerically by placing a regular grid over the region and using the nodes of the grid to approximate derivatives by adding and subtracting the values at certain grid points. The finite element method works with the

energy functional directly. The domain is divided into finite elements (regions of the 2D plane) which can be of any size or shape, and which are identified by their nodes which can be placed anywhere within the element and along its boundaries. Shape functions are defined over each element so that the values of the function inside the element are defined by its values at the nodes, and the problem is reduced to that of finding the values at these nodes. To do this, the original equation is reduced to a set of simultaneous equations which can be solved using Gaussian elimination.

We have chosen to use the finite element method for a number of reasons:

1. Since the elements in the finite element method can be of any shape it is much better for delineating curved boundaries. In addition the elements need not all be the same size so that the object can be covered with a finer mesh than the background. In this way the resolution of the elements can be decided by the problem.

2. Each element in the finite element mesh can have different material properties assigned to it. In this way the object of interest can be made stiffer than the background, for example.

3. The finite element method allows arbitrary constraints or boundary conditions such as fixed points or lines to be added or deleted simply.

The finite element method is used to warp each individual section and each warp is propagated to the whole stack by iteration. In fact there are three types of iteration: The finite element method requires iteration since we have large deformations and non-linear material properties. There is further iteration when warping the section because the forces change as the image of the section is warped. This means that the forces have to be recalculated after each finite element iteration. The final layer of iteration involves warping the whole stack. As each section is warped it influences the warp of its neighbouring sections. This problem is too large to solve in one go.

The details of the finite element method will be covered in chapter 6. However the finite element method requires the region to be covered by a mesh. We will now consider how this mesh can be obtained from the images automatically. Although elements of any

shape could be employed, we will just consider triangular elements as they are the simplest type of element. Triangles can delineate the boundaries of arbitrarily shaped regions, and thus can provide an adequate mesh for our application.

The finite element approximation approaches the true values as the mesh becomes finer, and more accurate results are obtained when the triangles are not too long and thin [107]. As we are not interested in how the background deforms, we require that the mesh is finer over the object than the background since fewer elements lead to less computation. We also require that the mesh delineates the different regions in the image in a reasonable way.

There are algorithms which will produce a triangulation from a set of points. Some of these methods produce a Delaunay triangulation which is a suitable triangulation to use because it maximises the minimum angle of the triangles. This prevents long thin triangles and makes the triangles as "regular" as possible. Another way of putting this is that a triangulation is a Delaunay triangulation if the circumcircle of each triangle contains only the three points of the triangle and no others. Therefore to triangulate the image, we will find a set of nodes and obtain a Delaunay triangulation using the "divide and conquer" technique of Lee and Schachter [61][1].

We now consider the calculation of the nodes. We would like the nodes to be located along the edges so that the boundaries on the triangles will follow the boundaries of the regions. These points should ideally be located at points of maximum curvature to fully describe the shapes of the regions. However we also require points in the middle of the regions to ensure a fine mesh. The points should be scattered more densely over the regions of the object than over the background to ensure that the mesh is finer over the object than the background.

The automatic triangulation algorithm employed in this project is:

1. Calculate the modulus of the gradient of the greylevels (here a derivative of a Gaussian of half-width at half-maximum of 1.0 pixels was used).

2. Automatically threshold the gradient image to separate the object from the back-

[1] I wish to thank Glynn Robinson (Department of Diagnostic Radiology, Yale University) for his implementation of this algorithm.

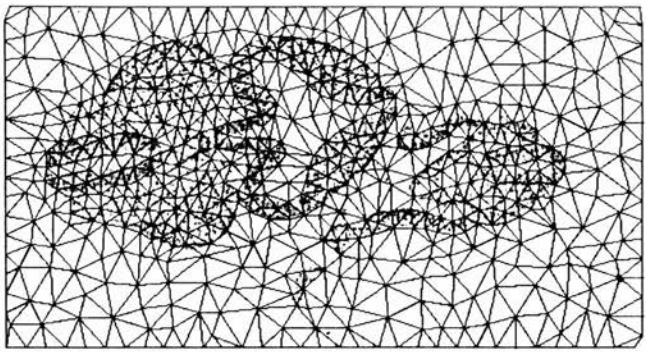


Figure 5.1: The finite element mesh for an image in the 9 day mouse embryo series.

ground[2]. To calculate the threshold the chord joining the histogram peak and the highest pixel value in the histogram was constructed. The histogram point furthest perpendicularly from this chord was chosen as the threshold. Here we are assuming that the histogram peak corresponds to the background values, and that background values are low while object pixel values are high.

3. All the points above threshold are ordered by gradient value, and starting from the highest value, points are chosen subject to not being too close to a point which has already been chosen. This produces a scatter of points over the object. Since the higher valued points are chosen first, the first points chosen should be located around the boundaries of regions.

4. To produce a scatter of points over the background, the rest of the points are ordered by gradient value. Points are chosen, again starting at the highest value, subject to being further than a second larger distance from all the other chosen nodes.

5. The Delaunay triangulation is calculated from the list of nodes.

Note that if the image is such that an equally fine mesh is required over the whole image, the two nodal separation distances can be equal.

[2] We shall see in chapter 7 that the object is characterised by high gradients while the background is characterised by low gradients.

This algorithm does not ensure that the nodes are at points of maximum curvature, and thus the boundaries of regions may not be followed as closely as we would like. However as we can see in figure 5.1, the results of this triangulation are largely satisfactory. A noticeably finer mesh covers the object, with a coarser mesh covering the background. We see that in most cases the edges of the mesh follow the boundaries of the regions, but in some cases this does not happen and an element containing parts of two regions is obtained. This is because choosing the points with the largest value first will not, in general, cause the nodes to be equally spaced around the edge. In fact spaces up to (but not including) twice the minimum separation distance may occur. Because of this a couple of nodes, one on each side of the edge, may be closer to each other and to the two edge nodes, than the edge nodes are to each other causing the Delaunay triangulation to join the two points on either side of the edge rather than the two edge points. The figure below illustrates this point. (a) shows the distribution of nodes along a line segment where two nodes are separated by a larger gap. Two nodes one in each region on either side of the line are also shown. (b) shows the Delaunay triangulation for this configuration, and (c) shows the triangulation which would be preferred. This

(a) (b) (c)

problem could possibly be solved by refining the mesh based on analysis of the greylevels in the elements, but this was not tried.

The finite element method can be used to warp the mesh, but we need a method of obtaining the warped image from the original image and the warped mesh. Using the nodal displacements and the interpolation functions, given a point in a particular element in the warped mesh we can calculate this point's position in the original mesh, and hence in the original image. However for this we need to know the element in which each point lies as we are scanning through the warped mesh. An efficient way of scanning the warped mesh while keeping track of the position of the points with respect to the mesh is the *plane sweep technique* [78].

This technique requires two data structures: the *event point schedule* and the *sweepline status*. The event point schedule is an ordered list of nodes each with a list of incoming

edges and a list of outgoing edges. The sweepline status describes the intersection of the sweepline with the triangulation. It consists of an ordered list of edges with enough information to calculate the warped image on one side of the edge. The sweepline is swept through the triangulation from the first node to the last node in the event point schedule. The sweepline pauses at each node to delete the incoming nodes and insert the outgoing nodes into the sweepline status. As the sweepline is moved over the mesh the warped image is calculated from information contained in the sweepline status.

The incoming and outgoing edges for each node can easily be calculated from a *doubly connected edge list* (DCEL), which can be obtained from the mesh. A DCEL consists of a list of edges with four information fields and two pointer fields. The information fields contain the vertices at each end of the edge and the faces on either side of the edge; the pointer fields point to the first edge encountered after the edge in question when one proceeds counterclockwise around the end vertices.

The plane sweep technique can also be used to assign different material properties to each element based on the greylevels in that element.

5.2.2 Calculating the Correspondences and Attaching the Springs

Finding corresponding points is very difficult with biological material and we cannot expect to be able to find exact correspondences all the time. To find correspondences we have to solve two problems: a suitable match function will have to be found, and correspondences have to be calculated from the resulting matchmap. A matchmap is the image obtained by matching a point in one image to a region of points in another image. The pixel values in the matchmap indicate how well the point matches each point in the other image. Note that because the sections are assumed to be registered we can assume that correspondences lie within small regions in the adjacent images, centred on the point to be matched. Because the sections are deformed we cannot pick the position of the best match as the corresponding point. For example the best match for a point on an edge will tend to move along the edge as the edge is deformed and the directions of parts of the edge vary. This is illustrated in figure 5.2 where a circle (a) has been deformed into an ellipse (b). Although points 1 and 2 match best to their corresponding points, point 3, which lies midway between points 1 and 2, does not, but

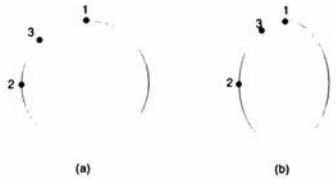


Figure 5.2: Deformation can cause the best match for a point to move along the edge as the edge is deformed. The circle in (a) has been deformed into an ellipse (b), and point 3 matches best to a point closer to point 1 than point 2 although its original position was midway between points 1 and 2.

instead matches best to a point closer to point 1 than point 2.

Since the position of the maximum value of the matchmap cannot be relied upon to indicate the correct correspondence we have to calculate an attraction to the areas in the matchmap where the point matches well so that a point in one of these areas can be picked as a corresponding point. To do this, we will either have to segment the matchmap into regions of good matches and bad matches, or sum over the whole matchmap to produce an attraction to the closest area of good match. To avoid the problems associated with segmentation and detailed analysis of the resulting image, we will take the latter approach. In this case we require high valued ridges on a low valued background so that only points which match well contribute significantly to the calculation. Further discussion of the problems of calculating correspondences from the matchmap will be found in chapter 8; chapter 7 discusses the matching problem.

Since it would be impractical to match every point in one image to every point in another image, and since there is little point in matching background points or points in areas of homogeneous texture, we need to determine some points which can be matched to adjacent images. These points should be in recognisable parts of the image, that is in areas of change such as boundaries between regions. We will call these points *interesting points*. These points should be independent of the finite element mesh as we may want more interesting points than nodes, or we may require that the interesting points are closer together, or further apart, than the nodes of the finite element mesh. We will need a sufficient number of interesting points so that incorrect correspondences do not greatly affect the warp. This implies that the interesting points should be scattered evenly over the image, especially over prominent features such as edges. However, if the

background is noisy with artefacts in adjacent sections which do not correspond, the accuracy of the warp may be affected as the warping process attempts to align these artefacts.

Since best matches of edge points will tend to slide along lines, we would expect to find corresponding lines rather than points for such edge points, and it would make sense to attach the springs to the corresponding lines in such a way that they are allowed to slide along the line with no penalty. There could be a problem with this in that if all the lines have the same direction, there will be no unique solution of the warp. However this will not be a problem as we are dealing with closed biological surfaces.

Since we cannot expect all the initial correspondences to be correct, we must allow the correspondences to change as the section is warped. We would also like the spring constants to reflect the likelihood that the correspondence is correct, and these may also change as the section is warped and features are brought into alignment.

In chapters 7 and 8 we will describe the methods used for matching and for calculating the correspondences and the strengths of the springs.

Chapter 6

The Finite Element Method.

6.1 Introduction

The finite element method was originally developed by engineers to solve complex structural problems such as finding the stresses and strains in an aeroplane wing. It is now widely used in engineering to solve all kinds of complex problems and mathematicians use it to find numerical solutions to partial differential equations and to minimise functionals.

The aim of the finite element method is to find a piecewise approximation to the exact solution. The approximate solution may be piecewise linear (C^1 continuous) or piecewise C^n continuous if a smoother function is required. The domain is split up into small pieces (such as triangles) characterised by their nodes, and appropriate interpolation functions (called shape functions) are defined over these regions (which are called elements). The shape functions are defined in terms of the values at the nodes. If the solution is required to be piecewise C^n continuous then the shape functions have to be C^{n-1} continuous at the boundaries of the elements. Because the values of the function inside the element are defined by its values at the nodes, the problem can be reduced to that of finding the values at the nodes. This is done by discretising the original equation to obtain a set of simultaneous equations which can be solved using, for example, Gaussian elimination.

This chapter describes how the finite element method can be used to solve the problem of using springs to warp an elastic plate. The finite element method was originally de-

veloped to solve problems in linear structural analysis where various assumptions can be made. In particular it is assumed that the displacements are sufficiently small that only small errors are introduced by the assumption that the original and final configurations are identical. This is clearly not the case when the aim is to warp an elastic plate. The problem of warping an elastic plate is non-linear and therefore non-linear techniques of finite element analysis have to be employed. The non-linear techniques involve approximating the non-linear equations by a set of linear equations. This approximate solution can then be refined by iteration where the equations are incrementally updated until the true solution has been obtained to the required accuracy.

Both linear and non-linear finite element solutions to the problem of warping an elastic plate will be described in detail. First an outline of the finite element method will be given. Then the appropriate equilibrium equations for use with the finite element equations will be derived. After a discussion of the characteristics of the finite element mesh and appropriate shape functions, and an introduction to linear elasticity, we will derive the finite element equations for linear analysis before proceeding to the description of a full non-linear analysis. Note that the following discussion consists mainly of a unification of the approaches of several text books [92, 8, 107, 108, 71] adapted to our particular application. The only difference between the method described here and that found in standard finite element text books is the method used to achieve convergence of non-linear analysis: the standard technique is to incrementally increase the load; we have chosen to limit the displacements at each iteration.

6.2 Outline of the Finite Element Method

The main stages of the application of the finite element method (FEM) are:

1. Determine a governing equilibrium equation for the problem under consideration.

2. Divide the domain into finite elements. These usually consist of polygons but other shapes can be used. The elements are defined by their nodes which are usually located at the vertices but can occur anywhere on the boundary or inside the element.

3. Choose the interpolating polynomials (or other functions) which will approximate the solution within each element. Continuity constraints can be satisfied by choosing appropriate functions. These functions should be defined in terms of the values at the nodes.

4. Evaluate the element stiffness matrix and force vector. Both the stiffness matrix and force vector are derived from the discretisation procedure of the functional, partial differential equation or virtual work equation, depending on the particular problem being analysed. In the case of elasticity the element stiffness matrix relates the forces to the nodal point displacements. The element force vector represents the external force (point force and distributed forces) acting at each node in the element.

5. Assemble the global stiffness matrix and force vector from the individual element stiffness matrices and force vectors. This produces a system of simultaneous equations which can be solved by Gaussian elimination or iterative methods.

These stages will now be described in more detail with regard to the problem of warping thin elastic plates. Notation will be introduced as it is needed, and a glossary of symbols is included in appendix B.

6.3 The Equilibrium Equation

We will derive the equilibrium equation for a single elastic plate with attached springs. We assume that initially the plate is not stressed but that the springs are under tension and exerting forces on the elastic plate. It is assumed that the springs have zero length at rest and that one end of each spring is attached to the plate while the other end is attached to a fixed point.

The total energy in this system is given by the sum of the internal energy in the plate and the internal energy in the springs. The internal energy in the plate is [81]

$$\frac{1}{2}\int_V \boldsymbol{\sigma}^T \boldsymbol{\epsilon}\, dV \tag{6.1}$$

where $\boldsymbol{\sigma}$ is the stress vector, $\boldsymbol{\epsilon}$ is the strain vector, and integration is over the current (warped) configuration.

The internal energy in a single spring at any point in the warping process is given by [82]

$$\frac{1}{2}\left(\mathbf{f}^i - \mathbf{d}^i\right)^T \mathcal{K}^i \left(\mathbf{f}^i - \mathbf{d}^i\right) \tag{6.2}$$

where $\mathbf{f}^i$ is the vector describing the initial extension of spring i, $\mathbf{d}^i$ is the vector describing the displacement of the point in the section to which the spring i is attached, and $\mathcal{K}^i$ is a 2×2 matrix containing the spring constant along the diagonal, and zeros elsewhere.

Clearly, the total energy contained in the springs is just the sum of the energy contained in each individual spring. Therefore the total energy, $\mathcal{E}$ in the system can be expressed (after multiplying out expression 6.2) as:

$$\mathcal{E} = \frac{1}{2}\int_V \boldsymbol{\sigma}^T \boldsymbol{\epsilon}\, dV + \sum\left(\frac{1}{2}\mathbf{f}^{iT}\mathcal{K}^i\mathbf{f}^i - \mathbf{d}^{iT}\mathcal{K}^i\mathbf{f}^i + \frac{1}{2}\mathbf{d}^{iT}\mathcal{K}^i\mathbf{d}^i\right) \tag{6.3}$$

where the sum is over all the springs attached to the plate. This is the governing energy functional which we will use for both linear and non-linear finite element analysis. The aim is to minimise this functional.

Note that although the above equation was derived by attaching the springs to points, in practice, as discussed in chapter 5, since we are matching edges, one end of each spring was attached to an infinite line. The lines are considered to be infinite because this is easy to include in the finite element method. As long as the corresponding lines do not all have the same direction, there will be a unique solution to the warping problem. This extension does not require major modifications to equation 6.3, the only changes being in the elements of the vector $\mathbf{f}^i$, and in the elements of $\mathcal{K}^i$ (which remains symmetric, but has rank 1). For details of these modifications see appendix C.

6.4 The Finite Element Mesh and Shape Functions

The first stage of the finite element method is to divide the domain into elements which are defined by the positions of their nodes. The elements need not all be the same shape. Elements are often polygons with straight sides, but arcs are possible. The nodes may be located anywhere inside the element or on the boundary, but they are usually located at the vertices. We will only consider triangular elements as they are the

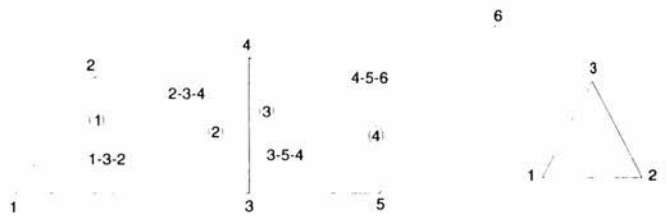


Figure 6.1: Global and local labelings; the local element labeling must proceed counter clockwise round the element.

simplest elements and adequate for the purposes of this problem. However the process of deriving the finite element equations can easily be applied to elements of any shape.

The nodes have both a global labeling and a local element labeling. For example in figure 6.1 the global labelings of the nodes of element 3 are the nodes 3, 4 and 5 where

node 3 corresponds to node 1 in the local labeling,
node 4 corresponds to node 3 in the local labeling,
node 5 corresponds to node 2 in the local labeling.

To distinguish between these two labelings, x_i and y_i will refer to the coordinates of node i in the global labeling, and x_i^e and y_i^e will refer to the node i in the local element labeling of element e. Note that the local element labeling must proceed counter clockwise round the element.

Interpolation functions are defined at each of the nodes. These functions are called shape functions and they relate the nodal displacements to the displacements inside the element. Here we will consider linear functions, but others are possible. It is important that the number of unknowns in the shape functions correspond to the number of degrees of freedom in the element. A typical *global shape function* $N_i(x, y)$ associated with node i has value zero over the whole finite element mesh except within the elements connected to node i. $N_i(x, y)$ has value one at node i and varies linearly (if it is a linear function) over all the elements attached to node i. The *element shape functions* N_i^e are the same as the global shape functions except that they are only defined over the

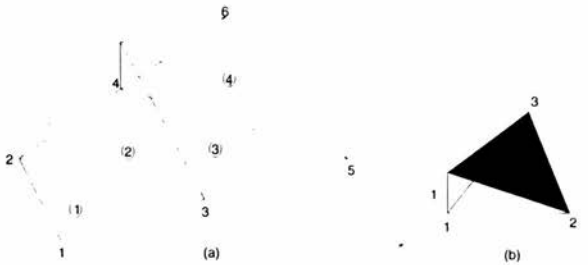


Figure 6.2: (a) Global shape function for node 4. (b) Local shape function for (local) node 1.

element in question and are numbered differently. For example, referring to figures 6.1 and 6.2 we see that

$$N_4 = N_3^2 + N_3^3 + N_1^4,$$

where

N_3^2 is the local shape function at local node 3 in element 2,
N_3^3 is the local shape function at local node 3 in element 3,
N_1^4 is the local shape function at local node 1 in element 4.

The displacements $\mathbf{u}^e$ $(= [u, v]^T)$ inside element e can be expressed as a linear function of the displacements at the nodes and the shape functions. Let the nodal displacements be given by $[u_1, v_1]^T$, $[u_2, v_2]^T$ and $[u_3, v_3]^T$ and let any point inside the element be given by $[x, y]^T$. Then

$$\begin{aligned} u(x, y) &= \sum N_i^e u_i \\ v(x, y) &= \sum N_i^e v_i \end{aligned}$$

which can be written in matrix form as

$$\mathbf{u}^e = \mathbf{N}^e \mathbf{a}^e, \tag{6.4}$$

where $\mathbf{N}^e$ is the element shape function matrix and $\mathbf{a}^e$ is the vector of nodal displacements for the element. For details on how to derive the element shape functions see, for example, Owen and Hinton [73] or Stasa [92].

6.5 Linear Analysis

The energy functional (equation 6.3) is defined in terms of the stresses and strains within the body. We are given the forces and wish to calculate the displacements, so we express both the stress and strain in terms of the displacements. In this section we will discuss stages four and five of the finite element method in detail, and for simplicity, we will assume linear and isotropic elasticity in the first instance. The elasticity is isotropic if the body exhibits the same properties in all directions and linear if the stress is proportional to the strain[1]. Non-linear elasticity will be considered in section 6.6.

6.5.1 Linear Elasticity

We first determine the stress and strain in a single element caused by the displacements within this element. The strain vector

$$\boldsymbol{\epsilon} = [\varepsilon_x, \varepsilon_y, \gamma_{xy}]^T$$

is defined in terms of $\mathbf{u}$ where

$$\varepsilon_x = \frac{\partial u}{\partial x} \tag{6.5}$$

and

$$\varepsilon_y = \frac{\partial v}{\partial y} \tag{6.6}$$

are the normal strains, and

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \tag{6.7}$$

is the shear strain (and relates to the rotation). In matrix form the strain can be written as

$$\boldsymbol{\epsilon} = \mathbf{L}\mathbf{u}^e,$$

[1] If the extension (strain) of a spring is proportional to the attached weight (stress) for all possible values of the weight, then the relationship is linear and known as Hooke's law.

where the superscript e indicates that the quantity is defined over element e, and

$$\mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix},$$

or

$$\boldsymbol{\epsilon} = \mathbf{B}^e \mathbf{a}^e, \tag{6.8}$$

where $\mathbf{a}^e$ us the vector of nodal displacements and

$$\mathbf{B}^e = \mathbf{L}\mathbf{N}^e,$$

where $\mathbf{N}^e$ is the element shape function matrix.

The stress vector $\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \tau_{xy}]^T$, where σ_x and σ_y are the normal stresses and τ_{xy} is the shear stress, is related to the strain vector by the stress-strain (or constitutive) matrix $\mathbf{D}$, with

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon}, \tag{6.9}$$

where for linear and isotropic elasticity

$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}.$$

Here E is the elastic modulus (or Young's modulus) and ν is Poisson's ratio. The elastic modulus relates the stress and the strain. It is a constant when the elasticity is linear. Poisson's ratio is the ratio of the increase in length in one direction to the decrease in length in a perpendicular direction. For isotropic materials (same properties in all directions), the ratios for both perpendicular directions are equal.

6.5.2 The Element Stiffness Matrix and the Force Vector

In this section we show how the finite element equations for a single element can be derived from the energy functional. For the moment we will make the assumption that the displacements are infinitesimal so that integration can be performed over the original configuration. In section 6.6 we will derive equations to deal with the case where this assumption is not made. The element stiffness matrix, which relates the nodal point

forces to the nodal point displacements, is derived from the energy functional. Applying the energy functional (equation 6.3) to a single element we obtain

$$\mathcal{E}^e = \frac{1}{2}\int_{V^e} \boldsymbol{\sigma}^T \boldsymbol{\epsilon}\, dV^e + \sum \left(\frac{1}{2}\mathbf{f}^{iT}\mathcal{K}^i\mathbf{f}^i - \mathbf{d}^{iT}\mathcal{K}^i\mathbf{f}^i + \frac{1}{2}\mathbf{d}^{iT}\mathcal{K}^i\mathbf{d}^i \right),$$

where the summation is over all the springs attached to the element. Here, the vector $\mathbf{d}^i$ represents the interpolated displacement of the point within the element to which the spring is attached. Normally in finite element analysis, the mesh is arranged so that the springs are attached to nodes; we will allow springs to be attached to any point within an element so that the finite element mesh can be independent of the springs.

Applying the relations

$$\begin{aligned} \boldsymbol{\epsilon} &= \mathbf{B}^e\mathbf{a}^e, \\ \boldsymbol{\sigma} &= \mathbf{D}\boldsymbol{\epsilon} \end{aligned}$$

given in the previous section to the above equation, and noting that $\mathbf{d}^i = \mathbf{N}^i\mathbf{a}^e$ where $\mathbf{N}^i$ is the shape function matrix given by the coordinates of spring i, which is attached to element e, we obtain

$$\mathcal{E}^e = \frac{1}{2}\int_{V^e} \mathbf{a}^{eT}\mathbf{B}^{eT}\mathbf{D}\mathbf{B}^e\mathbf{a}^e\, dV^e + \sum \left(\frac{1}{2}\mathbf{f}^{iT}\mathcal{K}^i\mathbf{f}^i - \mathbf{a}^{eT}\mathbf{N}^{iT}\mathcal{K}^i\mathbf{f}^i + \frac{1}{2}\mathbf{a}^{eT}\mathbf{N}^{iT}\mathcal{K}^i\mathbf{N}^i\mathbf{a}^e \right) \tag{6.10}$$

For an equilibrium we find the energy minimum using the calculus of variations [30]. The first variation is obtained by differentiating $\mathcal{E}^e$ with respect to $\mathbf{a}^e$ and multiplying by $\delta\mathbf{a}$. This is equivalent to finding $\mathcal{E}^e(\mathbf{a}^e + \delta\mathbf{a}) - \mathcal{E}^e(\mathbf{a}^e)$ where $\delta\mathbf{a}$ is arbitrary and small. The first variation of equation 6.10 is

$$\int_{V^e} \delta\mathbf{a}^T\mathbf{B}^{eT}\mathbf{D}\mathbf{B}\mathbf{a}^e\, dV^e - \sum \left(\delta\mathbf{a}^T\mathbf{N}^{iT}\mathcal{K}^i\mathbf{f}^i + \delta\mathbf{a}^T\mathbf{N}^{iT}\mathcal{K}^i\mathbf{N}^i\mathbf{a}^e \right) = 0.$$

$\delta\mathbf{a}$ is arbitrary and constant and therefore can be eliminated from the above equation giving

$$\left(\mathbf{K}^e + \mathbf{K}^e_{\mathbf{S}}\right)\mathbf{a}^e = \mathbf{F}^e_{\mathbf{I}} \tag{6.11}$$

where

$$\mathbf{K}^e = \int_{V^e} \mathbf{B}^{eT}\mathbf{D}\mathbf{B}^e\, dV^e, \tag{6.12}$$

$$\mathbf{K}^e_{\mathbf{S}} = \sum \mathbf{N}^{iT}\mathcal{K}^i\mathbf{N}^i, \tag{6.13}$$

$$\mathbf{F}^e_{\mathbf{I}} = \sum \mathbf{N}^{iT}\mathcal{K}^i\mathbf{f}^i. \tag{6.14}$$

The quantity $(\mathbf{K}^e + \mathbf{K}_\mathbf{S}^e)$ is known as the element stiffness matrix; $\mathbf{F}_\mathbf{I}^e$ is called the element force vector and represents the initial force exerted by the springs attached to the element.

Applying equation 6.12 to a triangular element we obtain

$$\mathbf{K}^e = h \int\int_{x,y} \mathbf{B}^{e^T} \mathbf{D} \mathbf{B}^e \, dx \, dy$$

where h is the thickness of the element. Since neither $\mathbf{B}^e$ nor $\mathbf{D}$ depend on x or y we obtain

$$\mathbf{K}^e = h \Delta \mathbf{B}^{e^T} \mathbf{D} \mathbf{B}^e$$

where Δ is the area of the triangle. Note that $\mathbf{K}^e$ is a symmetric matrix.

6.5.3 Assembly and Solution

The final stage in the finite element method is to combine all the information obtained by considering each element separately into a single set of simultaneous equations. This is done by assembling the element stiffness matrices and force vectors into a global stiffness matrix and force vector. The global stiffness matrix is obtained by adding global versions of the element stiffness matrices. The element stiffness matrices are converted into global versions by converting the local element labeling into the global labeling.

Since we have discretised the problem domain, we can evaluate the energy functional (equation 6.3) by summing the results of integrating the stress and strain over each element:

$$\mathcal{E} = \frac{1}{2} \sum_{\text{all elements}} \int_{V^e} \boldsymbol{\sigma}^T \boldsymbol{\epsilon} \, dV^e + \sum_{\text{all springs}} \left(\frac{1}{2} \mathbf{f}^{i^T} \boldsymbol{\lambda}^i \mathbf{f}^i - \mathbf{d}^{i^T} \boldsymbol{\lambda}^i \mathbf{f}^i + \frac{1}{2} \mathbf{d}^{i^T} \boldsymbol{\lambda}^i \mathbf{d}^i \right). \quad (6.15)$$

Referring to equation 6.11, the first variation of this can be written as

$$\sum_{\text{all elements}} \mathbf{K}^e \mathbf{a}^e + \sum_{\text{all springs}} \mathbf{K}_\mathbf{S}^e \mathbf{a}^e = \sum_{\text{all springs}} \mathbf{F}_\mathbf{I}^e.$$

Equation 6.4 can be interpreted as applying to the whole structure, that is

$$\mathbf{u} = \mathbf{N}\mathbf{a}$$

where **a** lists all the nodal displacements and

$$\mathbf{N} = \begin{cases} \mathbf{N}^e & \text{if the point concerned is in element } e \\ 0 & \text{otherwise} \end{cases}$$

Matrices **B**, $\mathbf{K_S}$, and vector $\mathbf{F_I}$ can be similarly defined and we can rewrite equations 6.11 and 6.12 as

$$(\mathbf{K} + \mathbf{K_S})\mathbf{a} = \mathbf{F_I} \tag{6.16}$$

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{D} \mathbf{B}\, dV \tag{6.17}$$

where **a** is the vector of nodal displacements, **K** is the global stiffness matrix, $\mathbf{F_I}$ is the global force vector and the integrals are taken over the (initial) whole volume of the object. Consider **K**, from the definitions of **B** and **N** we see that

$$\mathbf{K}_{ij} = \sum \int_{V^e} \mathbf{B}_l^{e\,T} \mathbf{D} \mathbf{B}_m^e\, dV^e$$

where node l in element e corresponds to node i in the global labeling, node m in element e corresponds to node j in the global labeling, and the summation is over all the elements containing the global nodes i and j. Note that $\mathbf{K}_{ij}$ is a $r \times r$ matrix where r is the number of degrees of freedom at each node (two in the case of thin elastic plates), and the local force vector is a $r \times 1$ matrix. The global stiffness matrix $(\mathbf{K} + \mathbf{K_S})$ is $rn \times rn$ and the global force vector is $rn \times 1$ where n is the number of nodes in the finite element mesh.

The assembly process is shown schematically in figure 6.3. The stars in the expanded element matrices represent $\mathbf{K}_{ij}$ in the element stiffness matrix. Note that the resulting stiffness matrix is sparse, banded and symmetric, and techniques for solving equations of this type have been developed [8]. The global stiffness matrix is sparse because each shape function is zero over most of the mesh. The labeling of the nodes is important as an inefficient labeling will increase the width of the bands which will in turn increase the amount of computation required to solve the simultaneous equations.

6.6 Non-Linear Finite Element Analysis

The linear analysis described above will give an approximate solution to the problem of using springs to warp an elastic plate but this approximation becomes increasingly

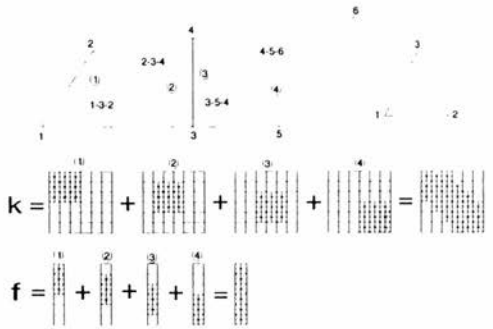


Figure 6.3: Schematic diagram for assembling the global stiffness matrix and the global force vector. Each square represents a (2×2) local stiffness matrix or a (2×1) local force vector.

inaccurate for large displacements. For a more accurate solution we turn our attention to non-linear methods.

Linear analysis contains several assumption:

1. The displacements are infinitesimal so that the geometry of the object does not change and integration can be performed over the original configuration.

2. The stress-strain relationship is constant; that is we are dealing with a linearly elastic material.

3. The strain-displacement relationship is independent of the displacement path.

If any of these assumptions are invalid we have to use non-linear analysis. For our purposes only assumption (3) is valid. Theoretically we could choose to model the plate with a material which behaves linearly for arbitrary large strain (although such a material does not exist), but in practice this can lead to numerical difficulties. Non-linear material properties will be discussed in detail in section 6.6.6.

The main reason why we cannot use the equations derived in section 6.5 is that, in the energy functional, the integration is over the stresses and strains in the final warped configuration of the plate. This configuration is unknown. Note also that, as the stress vector as defined above is not invariant to rigid body rotation, the above equations will

not deal satisfactorily with rigid body rotation of the plate or even of a single element.

These problems can be overcome by using alternative stress and strain measures which enable us to transform the stress and strain in the warped configuration to the original configuration and hence allow us to integrate over the original configuration. To define these new stress and strain measures and to derive the non-linear method of solution we will use the approach and a similar notation to that given by Bathe [8]. Since we will be dealing with non-linear equations for which matrix notation is not suitable we will use tensor notation in the following discussion.

6.6.1 Preliminary Discussion and Notation

As for linear analysis our starting point is the energy functional derived in section 6.3. For an equilibrium we need to minimise this energy and as before, we do this by setting the first variation to zero. The first variation of the energy functional (equation 6.3) is, remembering that $\mathcal{K}^i$ is symmetric,

$$\int_V \boldsymbol{\sigma}^T \delta\boldsymbol{\epsilon}\, dV + \sum \left(\mathbf{d}^{iT} \mathcal{K}^i \delta\mathbf{d}^i - \mathbf{f}^{iT} \mathcal{K}^i \delta\mathbf{d}^i \right) = 0 \tag{6.18}$$

where integration is over the warped configuration, and the sum is over all the springs attached to the plate. This equation is called the virtual work equation and represents the work done by infinitesimal displacements $\delta\mathbf{u}$ such that there is no change in the stresses in the plate or in the forces produced by the springs.

The main problem we have to solve is that of integrating over the unknown configuration of the warped plate and from this point we will assume that the plate is warped over time with the initial configuration at time 0 and the final configuration at time T. This will enable us to progress incrementally to the final solution should this be required. We will also assume that at time $t + \Delta t$, all the configurations up to and and including time t are known. Using this notation we can rewrite the virtual work equation as

$$\int_{t+\Delta t V} {}^{t+\Delta t}\boldsymbol{\sigma}^T \delta_{t+\Delta t}\boldsymbol{\epsilon}\ {}^{t+\Delta t}dV + \sum \left({}^{t+\Delta t}\mathbf{d}^{iT} \mathcal{K}^i \delta\mathbf{d}^i - \mathbf{f}^{iT} \mathcal{K}^i \delta\mathbf{d}^i \right) = 0 \tag{6.19}$$

The derivation of the finite element equations for the expression involving stress and strain is easier if tensor notation is used, as tensors allow us to express non-linear

relationships between the various quantities. In tensor notation the first expression in equation 6.19 is

$$\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}_{t+\Delta t}\sigma_{ij}\, \delta_{t+\Delta t}\epsilon_{ij}\, {}^{t+\Delta t}dV \tag{6.20}$$

where ${}^{t+\Delta t}_{t+\Delta t}\sigma_{ij}$ is the stress tensor at time $t+\Delta t$ and $\delta_{t+\Delta t}\epsilon_{ij}$ is the infinitesimal strain tensor referred to the configuration of the plate at time $t+\Delta t$. Using the notation given in section 6.5, the stress tensor σ_{ij} can be represented, in matrix form, as

$$\begin{bmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{bmatrix}$$

and the strain tensor ϵ_{ij} can be represented, in matrix form, as

$$\begin{bmatrix} \varepsilon_x & \frac{1}{2}\gamma_{xy} \\ \frac{1}{2}\gamma_{xy} & \varepsilon_y \end{bmatrix}.$$

Note that ϵ_{ij} can also be written (in tensor notation) as

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

where $u_1 = u(x,y)$ and $u_2 = v(x,y)$ are the displacement functions as defined in section 6.4. Similarly $x_1 = x$ and $x_2 = y$ in the usual cartesian notation.

We now formally define the notation which we will use. The coordinates of a point P in the plate at time 0 are ${}^0x_1$ and ${}^0x_2$; at time $t+\Delta t$ they are ${}^{t+\Delta t}x_1$ and ${}^{t+\Delta t}x_2$, where the left superscripts refer to the configuration of the body and the right subscripts to the coordinate axes. The notation for the displacements of the plate is similar: at time t the displacements are ${}^tu_i$, $i = 1,2$ and we have ${}^tx_i = {}^0x_i + {}^tu_i$, $i = 1,2$. The increments in the displacements from time t to time $t+\Delta t$ are

$$u_i = {}^{t+\Delta t}u_i - {}^tu_i.$$

Since the configuration of the plate at time $t+\Delta t$ is unknown, we will refer applied stresses and strains to a known configuration. In analogy to the notation used for coordinates and displacements, a left superscript indicates the configuration in which the stress or strain occurs; a left subscript indicates the configuration with respect to which the stress or strain is measured. If the quantity under consideration occurs in the same configuration with respect to which it is measured, the left subscript may be omitted.

As in the linear case we will also have to consider derivatives of displacements and coordinates. For these we will use the standard tensor notation in that a comma will denote differentiation with respect to the coordinate following the comma; and the left subscript, denoting time will indicate the configuration in which this coordinate is measured. For example

$$ {}^t_0 u_{i,j} = \frac{\partial {}^t u_i}{\partial {}^0 x_j} $$

6.6.2 The 2nd Piola-Kirchhoff Stress Tensor and the Green-Lagrange Strain Tensor

From now on the stress and strain measures defined in section 6.5 will be referred to as *engineering* stress and strain. We define new stress and strain measures which will enable us to refer the stresses and strains in an unknown configuration to a known configuration.

The 2nd Piola Kirchhoff stress tensor at time t, referred to the configuration at time 0 is defined as

$$ {}^t_0 S_{ij} = \frac{{}^0\rho}{{}^t\rho} {}^t_0 x_{i,m} {}^t_t \sigma_{mn} {}^t_0 x_{j,n} \tag{6.21} $$

where ${}^t\rho$ is the mass density at time t, ${}^t_t\sigma_{mn}$ is the engineering stress at time t (referred to configuration t), and ${}^t_0 x_{i,m}$ is a member of the deformation tensor, represented by the matrix

$$ {}^t_0\mathbf{X} = \begin{bmatrix} \frac{\partial {}^t x_1}{\partial {}^0 x_1} & \frac{\partial {}^t x_1}{\partial {}^0 x_2} \\ \frac{\partial {}^t x_2}{\partial {}^0 x_1} & \frac{\partial {}^t x_2}{\partial {}^0 x_2} \end{bmatrix} $$

The mass density ratio can be evaluated since the mass of the plate is conserved under warping, and it can be shown [8] that

$$ {}^0\rho = {}^t\rho \det {}^t_0\mathbf{X}. $$

The 2nd Piola Kirchhoff stress tensor is symmetric and invariant to rigid body motion, unlike the engineering stress and strain tensors.

The strain tensor used with the 2nd Piola-Kirchhoff stress tensor is the Green-Lagrange strain tensor. The Green-Lagrange strain tensor at time t, referred to the configuration at time 0 is defined as

$$ {}^t_0 G_{ij} = \tfrac{1}{2}({}^t_0 u_{i,j} + {}^t_0 u_{j,i} + {}^t_0 u_{k,i} {}^t_0 u_{k,j}) \tag{6.22} $$

The Green-Lagrange strain tensor is symmetric and invariant to rigid body transformations.

It can be shown that [8]

$$\int_{{}_0V} {}^t_0S_{ij}\, \delta {}^t_0G_{ij}\, {}^0dV = \int_{{}_tV} {}^t_t\sigma_{mn}\, \delta {}_t\epsilon_{mn}\, {}^tdV. \tag{6.23}$$

Therefore, we see that the 2nd Piola-Kirchhoff stress tensor and the Green-Lagrange strain tensor transform the engineering stress and strain measures to the original configuration.

Note that the 2nd Piola-Kirchhoff stress tensor and the Green-Lagrange strain tensor have little physical meaning, but for problems involving rigid body movements and infinitesimal strain they provide natural stress and strain measures because only the actual distortion of the object contributes to the stress and strain. For this reason, any material description which has been developed for small strain analysis using engineering stress and strain measures can be used in this type of problem provided the 2nd Piola-Kirchhoff stresses and Green-Lagrange strains are used. The situation with problems involving large strain is more problematic. For these types of problems it is essential that appropriate material descriptions are used, otherwise the results will only be valid for small deformations.

6.6.3 Derivation of the Finite Element Governing Equations

Now that we have defined the relevant stress and strain measures we are in a position to derive the finite element governing equations for non-linear analysis. For the moment we will only consider the expressions involving the energy in the elastic plate as this is the non-linear part of the energy functional. Substituting the expression involving the 2nd Piola-Kirchhoff stress and Green-Lagrange strain, from equation 6.23 for $t = t + \Delta t$ into expression 6.20 we obtain

$$\int_{{}_0V} {}^{t+\Delta t}_0S_{ij}\, \delta {}^{t+\Delta t}_0G_{ij}\, {}^0dV. \tag{6.24}$$

We notice that ${}^t_0G_{ij}$ is a non-linear function of the displacements. Therefore we need to linearise expression 6.24 so that it can be applied to the finite element formulation. The linear approximations necessitate an iterative solution and an incremental approach

involving several time steps may be necessary. For the following we assume that the solutions for time 0 to time t have been found and we seek a solution for time $t + \Delta t$.

Since the configuration at time $t + \Delta t$ is unknown, we refer the applied stresses and strains to a known configuration. There are two possibilities: we could either refer to the configuration at time t (Updated Lagrangian (UL) formulation) or the configuration at time 0 (Total Lagrangian (TL) formulation). As long as appropriate constitutive relations (different for each formulation) are used, both formulations will produce identical results. However the TL formulation is normally used for analysis of solids and structures whereas the UL formulation is normally used for fluid mechanics problems. For this reason we will employ the TL formulation.

Since all the configurations up to time t are known we can decompose the stresses and strains at time $t + \Delta t$ as follows.

The stresses can be decomposed as

$$ {}^{t+\Delta t}_{0}S_{ij} = {}^{t}_{0}S_{ij} + {}_{0}S_{ij} $$

where ${}_{0}S_{ij}$ is the incremental 2nd Piola-Kirchhoff stress.

We can decompose the strains as

$$ {}^{t+\Delta t}_{0}G_{ij} = {}^{t}_{0}G_{ij} + {}_{0}G_{ij}, $$

where ${}_{0}G_{ij}$ is the incremental Green-Lagrange strain. It is easy to show that

$$ {}_{0}G_{ij} = {}_{0}e_{ij} + {}_{0}\eta_{ij}, $$

where

$$ {}_{0}\eta_{ij} = \frac{1}{2}({}_{0}u_{k,i}\,{}_{0}u_{k,j}) \tag{6.25} $$

$$ {}_{0}e_{ij} = \frac{1}{2}({}_{0}u_{i,j} + {}_{0}u_{j,i} + {}^{t}_{0}u_{k,i}\,{}_{0}u_{k,j} + {}_{0}u_{k,i}\,{}^{t}_{0}u_{k,j}) \tag{6.26} $$

The last two terms for ${}_{0}e_{ij}$ are due to an initial displacement effect; ${}_{0}\eta_{ij}$ and ${}_{0}e_{ij}$ are the non-linear and linear incremental strains referred to the configuration at time 0.

Substituting these into expression 6.24 and noting that $\delta^{t+\Delta t}_{0}G_{ij} = \delta_{0}G_{ij}$, we obtain

$$ \int_{{}_{0}V} {}_{0}S_{ij}\,\delta_{0}G_{ij}\,{}^{0}dV + \int_{{}_{0}V} {}^{t}_{0}S_{ij}\,\delta_{0}\eta_{ij}\,{}^{0}dV + \int_{{}_{0}V} {}^{t}_{0}S_{ij}\,\delta_{0}e_{ij}\,{}^{0}dV \tag{6.27} $$

for the virtual work done in the plate.

This expression can be linearised by making the approximations

$$
\begin{aligned}
{}_0S_{ij} &= {}_0C_{ijrs}\,{}_0e_{rs} \\
\delta_0 G_{ij} &= \delta_0 e_{ij}
\end{aligned}
$$

where ${}_0C_{ijrs}$ is the incremental property tensor at time t referred to the configuration at time 0.

Substituting these approximations into expression 6.27 we obtain

$$
\int_{{}_0V} {}_0C_{ijrs}\,{}_0e_{rs}\,\delta_0 e_{ij}\,{}^0dV + \int_{{}_0V} {}^t_0S_{ij}\,\delta_0\eta_{ij}\,{}^0dV + \int_{{}_0V} {}^t_0S_{ij}\,\delta_0 e_{ij}\,{}^0dV \qquad (6.28)
$$

Finally substituting this expression into the virtual work equation 6.19, and writing the displacements ${}^{t+\Delta t}\mathbf{d}^i$ at time $t+\Delta t$ as

$$
{}^{t+\Delta t}\mathbf{d}^i = {}^t\mathbf{d}^i + \Delta\mathbf{d}^i.
$$

where $\Delta\mathbf{d}$ gives the incremental displacements, we obtain (mixing tensor and matrix notation)

$$
\begin{aligned}
\int_{{}_0V} {}_0C_{ijrs}\,{}_0e_{rs}\,\delta_0 e_{ij}\,{}^0dV + \int_{{}_0V} {}^t_0S_{ij}\,\delta_0\eta_{ij}\,{}^0dV + \sum \Delta\mathbf{d}^{iT}\mathcal{K}^i\delta\mathbf{d}^i = \\
\sum \left(\mathbf{f}^{iT}\mathcal{K}^i\delta\mathbf{d}^i - {}^t\mathbf{d}^{iT}\mathcal{K}^i\delta\mathbf{d}^i\right) - \int_{{}_0V} {}^t_0S_{ij}\,\delta_0 e_{ij}\,{}^0dV \qquad (6.29)
\end{aligned}
$$

Once the configuration for time t is known we can use this equation to calculate an increment in the displacements. These displacements can then be used to evaluate approximations to the total displacements, strains (use equation 6.22 for the Green-Lagrange strain and equation 6.8 for the engineering strains), and the stresses (use the constitutive relations given in section 6.6.6) corresponding to time $t+\Delta t$. Once these approximate displacements, strains, and stresses have been calculated, they can be used to calculate the difference between the internal energy in the plate and the internal energy in the springs.

6.6.4 Discretisation

There are three expressions which must be discretised; the expressions involving springs will be put into terms similar to those derived in section 6.5.

$$\int_{{}_0V} {}_0C_{ijrs}\, {}_0e_{rs}\, \delta {}_0e_{ij}\, {}^0dV$$
$$\int_{{}_0V} {}^t_0S_{ij}\, \delta {}_0\eta_{ij}\, {}^0dV$$
$$\int_{{}_0V} {}^t_0S_{ij}\, \delta {}_0e_{ij}\, {}^0dV$$

These expressions can be represented in matrix form as [8]

$$\int_{{}_0V} \mathbf{a}^T\, {}^t_0\mathbf{B}_\mathbf{L}^T\, {}_0\mathbf{C}\, {}^t_0\mathbf{B}_\mathbf{L}\, \delta\mathbf{a}\, {}^0dV \tag{6.30}$$
$$\int_{{}_0V} \mathbf{a}^T\, {}^t_0\mathbf{B}_\mathbf{NL}^T\, {}^t_0\mathbf{S}\, {}^t_0\mathbf{B}_\mathbf{NL}\, \delta\mathbf{a}\, {}^0dV \tag{6.31}$$
$$\int_{{}_0V} {}^t_0\mathbf{B}_\mathbf{L}^T\, {}^t_0\hat{\mathbf{S}}\, \delta\mathbf{a}\, {}^0dV \tag{6.32}$$

respectively, where ${}_0\mathbf{C}$ is the incremental constitutive matrix, ${}^t_0\mathbf{B_L}$ is the linear strain displacement matrix, ${}^t_0\hat{\mathbf{S}}$ is the vector corresponding to the 2nd Piola-Kirchhoff stress tensor at time t referred to configuration 0:

$${}^t_0\hat{\mathbf{S}} = [{}^t_0S_{11},\ {}^t_0S_{22},\ {}^t_0S_{12}],$$

and ${}^t_0\mathbf{B_{NL}}$ is the non-linear strain-displacement transformation matrix.

Substituting expressions 6.30, 6.31, and 6.32 in equation 6.29, and using the relation ${}^t\mathbf{d}^i = \mathbf{N}^i\, {}^t\mathbf{a}$ we obtain

$$\int_{{}_0V} \mathbf{a}^T\, {}^t_0\mathbf{B}_\mathbf{L}^T\, {}_0\mathbf{C}\, {}^t_0\mathbf{B}_\mathbf{L}\, \delta\mathbf{a}\, {}^0dV + \int_{{}_0V} \mathbf{a}^T\, {}^t_0\mathbf{B}_\mathbf{NL}^T\, {}^t_0\mathbf{S}\, {}^t_0\mathbf{B}_\mathbf{NL}\, \delta\mathbf{a}\, {}^0dV + \sum \mathbf{a}^T \mathbf{N}^{iT} \mathcal{K}^i \mathbf{N}^i \delta\mathbf{a} =$$
$$\sum \mathbf{f}^{iT} \mathbf{N}^{iT} \mathcal{K}^i \mathbf{N}^i \delta\mathbf{a} - \sum {}^t\mathbf{a}^T \mathbf{N}^{iT} \mathcal{K}^i \mathbf{N}^i \delta\mathbf{a} - \int_{{}_0V} {}^t_0\mathbf{B}_\mathbf{L}^T\, {}^t_0\hat{\mathbf{S}}\, \delta\mathbf{a}\, {}^0dV$$

where ${}^t\mathbf{a}$ is the vector of nodal displacements at time t, and $\mathbf{a}$ is the vector of incremental nodal displacements. We note that $\delta\mathbf{a}$ is arbitrary and can thus be eliminated from this equation to give

$$({}^t_0\mathbf{K_L} + {}^t_0\mathbf{K_{NL}} + \mathbf{K_S})\mathbf{a} = \mathbf{F_I} - {}^t_0\mathbf{F_S} - {}^t_0\mathbf{F_P}. \tag{6.33}$$

where

$${}^t_0\mathbf{K}_\mathbf{L} = \int_{{}_0V} {}^t_0\mathbf{B}_\mathbf{L}^T\, {}_0\mathbf{C}\, {}^t_0\mathbf{B}_\mathbf{L}\, {}^0dV$$

$$
\begin{aligned}
{}^{t}_{0}\mathbf{K}_{\mathbf{NL}} &= \int_{{}_0V} {}^{t}_{0}\mathbf{B}^{T}_{\mathbf{NL}}\, {}^{t}_{0}\mathbf{S}\, {}^{t}_{0}\mathbf{B}_{\mathbf{NL}}\, {}^{0}dV \\
{}^{t}_{0}\mathbf{F}_{\mathbf{S}} &= \sum {}^{t}\mathbf{a}^{T}\mathbf{N}^{i\,T}\mathcal{K}_{i}\mathbf{N}^{i} \\
{}^{t}_{0}\mathbf{F}_{\mathbf{P}} &= \int_{{}_0V} {}^{t}_{0}\mathbf{B}_{\mathbf{L}}\, {}^{t}_{0}\hat{\mathbf{S}}\, {}^{0}dV,
\end{aligned}
$$

and $\mathbf{K_S}$ and $\mathbf{F_I}$ are defined as before. Note that ${}^{t}_{0}\mathbf{K}_{\mathbf{L}}$, ${}^{t}_{0}\mathbf{K}_{\mathbf{NL}}$, and $\mathbf{K_S}$ are all symmetric.

6.6.5 Iteration

We can now use equation 6.33 to iterate towards the solution. For the moment we will assume that the applied forces (that is the springs) do not change. For iteration we rewrite equation 6.33 as

$$
({}^{t}_{0}\mathbf{K}_{\mathbf{L}} + {}^{t}_{0}\mathbf{K}_{\mathbf{NL}} + \mathbf{K_S})\Delta\mathbf{A}^{(i)} = \mathbf{F_I} - {}^{t+\Delta t}_{0}\mathbf{F}_{\mathbf{S}}{}^{(i-1)} - {}^{t+\Delta t}_{0}\mathbf{F}^{(i-1)}_{\mathbf{P}}. \tag{6.34}
$$

where $\Delta\mathbf{A}^{(i)}$ is the vector of increments in the nodal point displacements in iteration i. The initial conditions for the iteration are

$$
\begin{aligned}
\Delta\mathbf{A}^{(0)} &= {}^{t}\mathbf{a} \\
{}^{t+\Delta t}_{0}\mathbf{F}^{(0)}_{\mathbf{S}} &= {}^{t}_{0}\mathbf{F}_{\mathbf{S}} \\
{}^{t+\Delta t}_{0}\mathbf{F}^{(0)}_{\mathbf{P}} &= {}^{t}_{0}\mathbf{F}_{\mathbf{P}}\,.
\end{aligned}
$$

For updating we have

$${}^{t+\Delta t}\mathbf{a}^{(i)} = {}^{t+\Delta t}\mathbf{a}^{(i-1)} + \Delta\mathbf{A}^{(i)}$$

which can be used to update the stress and strain in the plate and the springs, and hence obtain new approximations to ${}^{t+\Delta t}_{0}\mathbf{F}_{\mathbf{S}}$ and ${}^{t+\Delta t}_{0}\mathbf{F}_{\mathbf{P}}$. We continue this iteration until $\Delta\mathbf{A}^{(i)}$ is sufficiently small.

This iteration procedure will converge only when the initial (linear) approximation is close enough to the real solution. It is to allow the process to converge that the calculation is split into several time intervals. The general engineering approach is to calculate the response of the system under a variety of loadings. In this case the procedure is to find the solution for a small loading and then to use this result to calculate the solution for a slightly larger loading. This loading iteration is continued

until the final solution is reached. In the application of warping serial sections, we are only interested in the solution for the final loading.

Since stronger forces should result in larger displacements, another method of ensuring convergence is to limit the allowed nodal displacements in a single iteration. This limit should clearly be relative to the size of the element. One measure which is easy to calculate is the quotient of the maximum nodal displacement of the element and the area of the element. This measure has been used successfully to enable the finite element calculations to converge[2].

A comparison of using linear and non-linear methods is shown in figure 6.4. Two springs were attached to opposite corners of a 64×64 square of material 10 units thick, with a Poisson's ratio $\mu = 0.0$ and an elastic modulus $E = 5.0$. The spring constants were set to 100 and the initial state of the system is shown in part (a) of figure 6.4. The mesh was divided into 2×16^2 regular triangles and the results of applying both linear and non-linear methods of solution to this problem are shown in parts (b) and (c) of figure 6.4. In part (c) the system is in equilibrium, but this is not the case in part (b). We notice that the purely linear solution tends to overestimate the displacements.

We now turn our attention to the case where the applied forces change as the plate is warped. This is equivalent to warping the plate a little and then changing the springs while keeping the plate under tension. This can easily be built into the above finite element iteration scheme by adding another layer of iteration, which we will call force iteration. First the spring forces and stiffness matrices are calculated relative to the original configuration and the finite element iteration is employed to bring the system into equilibrium. Then the spring forces and spring stiffness matrix are calculated relative to the new configuration, but the section stiffness matrix and the stresses and strains in the plate are calculated relative to the original configuration. This force iteration is continued until the displacement between successive force iterations is small. The forces represented by the springs must be calculated consistently between force iterations otherwise the process will not converge.

[2] It was realised at a late stage that this parameter should be dimensionless and that it should therefore be the quotient of the *square* of the maximum nodal displacement and the area of the element. However in cases where the finite element iteration converges, this will make no difference to the results.

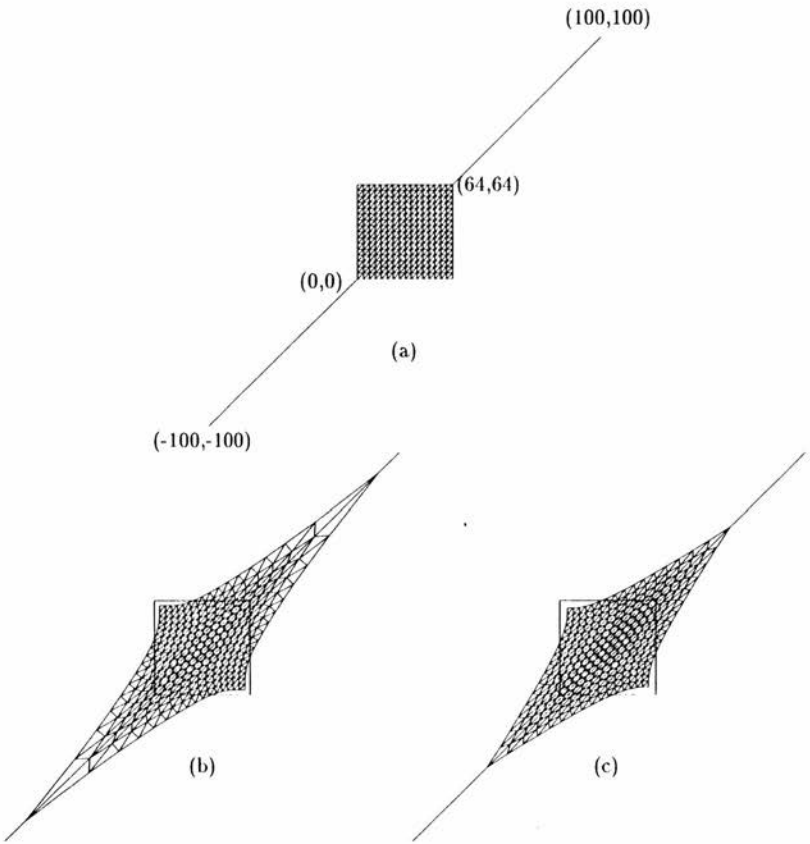


Figure 6.4: (a) Two springs were attached to a 64×64 square of material. (b) The linear solution; the system is not in equilibrium. (c) The non-linear solution; 41 iterations were needed to bring the system into equilibrium.

Note that to ensure convergence, the finite element calculations must be allowed to converge to sufficient accuracy so that the system is in equilibrium for the particular set of forces. Then when the forces are refined, the warp will also be refined. If the system is not in equilibrium when the forces are updated, there is little chance that the new forces will cause the system to be in equilibrium. In fact, updating the forces may introduce instability because the displacements are overestimated if only a single finite element iteration is employed, causing over correction for the change in the forces.

It was found that attaching the springs to infinite lines also caused convergence problems, possibly due to numerical instability. This problem was solved by attaching each spring to the nearest point on its corresponding line, and updating the attachment point at each finite element iteration. In this way the points are allowed to slide freely along their corresponding lines as the finite element iteration progresses, but are restricted in their movement during each iteration.

6.6.6 Non-Linear Material Properties

We now consider the problem of determining appropriate constitutive relations for use with non linear finite element analysis. First the consequences of using linear material properties will be discussed and then we will consider the formulation of material properties for hyper-elastic materials.

If we use the linear constitutive relations given in section 6.5, the UL and TL formulations do not produce identical results. For example consider the problem of calculating the displacement undergone by a unit cube fixed at one face with a force acting on the opposite face. The cube is resting on a smooth surface. If we solve this problem analytically using the normal linear approach and the UL and TL formulations, described in section 6.6.3, using the stress and strain tensors defined in section 6.6.2, and plot the force required for a given displacement we obtain the graph in figure 6.5. For these calculations we used the following values for the material constants: $E = 2$ and $\nu = 0$. From this graph we see that the UL and TL formulations do not produce a linear stress strain relationship when a linear stress strain relationship is used; however, a linear response is seen for small displacements (up to 0.1). The reason for this is that although the stress strain relationship is linear, the strain displacement relationship is

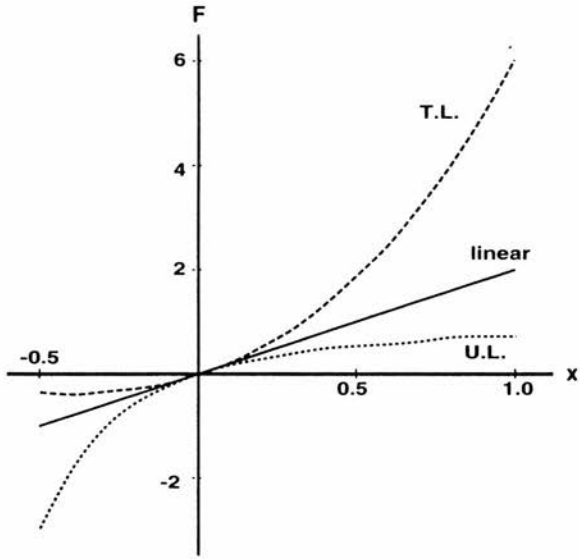


Figure 6.5: Graph showing force (F) needed to cause a displacement x of a unit cube under tension on one face and fixed at the opposite face.

non-linear resulting in a non-linear stress-displacement relationship.

From figure 6.5 we see that neither stress-displacement curve is suitable for large displacement analysis since the TL formulation will cope well with stretching, but not compression; and the UL formulation will cope well with compression but not stretching. For this reason we will now consider non-linear constitutive relationships. In particular we will consider the properties of hyper-elastic materials. Note that the details of the following discussion can be found in the book by R. W. Ogden [71].

The mechanical properties of a hyperelastic (or Green elastic) material are characterised by the strain energy function W where

$$\underline{\mathbf{S}} = \frac{\partial W(\underline{\mathbf{G}})}{\partial \underline{\mathbf{G}}}$$

Here $\underline{\mathbf{S}}$ is the 2nd Piola-Kirchoff stress tensor and $\underline{\mathbf{G}}$ is the Green-Lagrange strain tensor. It can be shown that [71] for an isotropic material, W is an isotropic scalar function which may be regarded as a symmetric function of the principal stretches λ_i. The principal stretches are real numbers greater than zero which indicate the deformation

in three orthogonal directions[3]. It can be shown that if g_i are the eigenvalues of the Green-Lagrange strain tensor then the principal stretches are given by

$$\lambda_i = \sqrt{(2g_i + 1)}.$$

This leads to the following form for the strain energy function:

$$W(\lambda_1, \lambda_2, \lambda_3) = \sum_{p=1}^{N} (u_p(\lambda_1^{\alpha_p} + \lambda_2^{\alpha_p} + \lambda_3^{\alpha_p})/\alpha_p + g(J)),$$

where $J = \lambda_1\lambda_2\lambda_3$ is the local ratio of current to reference volume. For an incompressible material (where $J = 1$) the simplest strain energy function is

$$W(\lambda_1, \lambda_2) = u_1(\lambda_1^{\alpha_1} + \lambda_2^{\alpha_1} + \lambda_1^{-\alpha_1}\lambda_2^{-\alpha_1} - 3)/\alpha_1 + u_2(\lambda_1^{\alpha_2} + \lambda_2^{\alpha_2} + \lambda_1^{-\alpha_2}\lambda_2^{-\alpha_2} - 3)/\alpha_2,$$

and for a compressible material the simplest strain energy function (corresponding to a Poisson's ratio of 0.0) is

$$W(\lambda_1, \lambda_2) = u_1(\lambda_1^{\alpha_1} + \lambda_2^{\alpha_1} - 2)/\alpha_1 + u_2(\lambda_1^{\alpha_2} + \lambda_2^{\alpha_2} - 2)/\alpha_2,$$

noting that for the problem of warping thin elastic plates $\lambda_3 = 1$.

Given such a strain energy function, the principal components of the 2nd Piola Kirchoff stress tensor are given by

$$s_i = \frac{\partial W}{\partial g_i}$$

for a compressible material, and

$$s_i = \frac{\partial W}{\partial g_i} - p\lambda_i^2$$

for an incompressible material, where p is the (arbitrary) hydrostatic stress. Note that for problems involving a thin elastic plate where no forces are acting through the thickness of the plate, p is not arbitrary and is

$$p = \lambda_1^{-2}\lambda_2^2\frac{\partial W}{\partial g_3}.$$

Given the principal components, the 2nd Piola Kirchoff stress tensor is given by

$$\mathbf{S} = \sum s_i \mathbf{g}^{(i)} \otimes \mathbf{g}^{(i)},$$

[3] The principal stretches are actually defined with respect to the deformation tensor. The deformation tensor $\mathbf{X}$ can be decomposed in a rotational component $\mathbf{R}$ and a deformation component $\mathbf{U}$ such that $\mathbf{X} = \mathbf{RU}$. Then the principal stretches are the eigenvalues of $\mathbf{U}$.

where $\mathbf{g}^{(i)}$ are the eigenvectors of the Green-Lagrange strain tensor and $\otimes$ denotes the tensor product ($(\mathbf{u} \otimes \mathbf{v})_{ij} = u_i v_j$).

We now consider expressions for the incremental material property tensor ${}_0\underline{\mathbf{C}}$. For a linear stress-strain relationship ${}_0\underline{\mathbf{C}}$ is constant and is equal to the constitutive matrix $\mathbf{D}$ in section 6.5; for a non-linear stress-strain relationship the incremental material property tensor should reflect the slope of the stress-strain curve at the appropriate point. Therefore

$$ {}_0\underline{\mathbf{C}} = \frac{\partial \mathbf{S}}{\partial \underline{\mathbf{G}}}. $$

It can be shown that the non-zero components of ${}_0\underline{\mathbf{C}}$ are given by

$$ \begin{aligned} {}_0C_{iijj} &= \frac{\partial s_i}{\partial g_j} \\ {}_0C_{ijij} &= \tfrac{1}{2}\frac{s_i - s_j}{g_i - g_j} \quad i \neq j \end{aligned} $$

subject to the symmetries ${}_0C_{ijkl} = {}_0C_{jikl} = {}_0C_{ijlk}$. However this assumes that $g_1 \neq g_2 \neq g_3 \neq g_1$. If $g_i = g_j \neq g_k$ the non-zero components are

$$ \begin{aligned} {}_0C_{iijj} &= \frac{\partial s_i}{\partial g_j} \\ {}_0C_{ijij} &= \tfrac{1}{2}\frac{s_i - s_j}{g_i} \quad i \neq j, \end{aligned} $$

and if $g_1 = g_2 = g_3$ we have

$$ {}_0C_{ijkl} = \beta_1 \delta_{ij}\delta_{kl} + 2\beta_2(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), $$

where

$$ \begin{aligned} \beta_1 &= \frac{\partial s_i}{\partial g_j} \\ \beta_1 + 2\beta_2 &= \frac{\partial s_i}{\partial g_i}, \end{aligned} $$

and δij is the Kronecker delta. Note that for an incompressible material $\beta_1 = 0$.

6.7 Conclusion

In this chapter we have described how the finite element method can be used to warp a thin elastic plate. Finite element equations have been given to solve problems involving large deformations, and hyperelastic material properties. The discussion has been

focused on the problem of *plane stress*, that is problems involving a thin elastic plate where all movement is within the plane of the plate, but the finite element method can, and has been, applied to other problems in structural analysis including bending of bars and plates, and full 3D warping. For details of applying the finite element method to these problems see, for example, the books by Bathe [8] and Zienkiewicz and Taylor [107, 108].

In the following chapters we will discuss the calculation of the forces to be applied to the sections, and show the results of using the finite element method to warp serial section images.

Chapter 7

Matching Adjacent Serial Section Images

7.1 Introduction

In order to correct the sectioning distortions we model each section as a thin elastic plate and apply forces (modelled as springs) to the section images arising from matched image regions. In this chapter we consider the matching problem. We assume that the images are registered and that there is no correlation in the distortion of adjacent sections.

Most correspondence algorithms involve two stages: first a match function is used to calculate how well each individual feature (generally a point, line or region) in one image matches each of a number of similar features in the other image, then a global cost function is used to assign each feature to its corresponding feature. In this chapter we will only consider the match function and the problem of finding suitable features to match; the problem of finding correspondences will be discussed in chapter 8. There are two parts to finding a good match function. We must first decide on the type of feature we would like to match and determine attributes associated with these features. Then we must determine a match function based on these attributes. The attributes may include, for a point, the gradient magnitude and direction at that point, or, for a line segment, the length and direction of the line segment.

We show that approaches based on gradient information do not meet the criteria for our particular application, and then we describe a new match function designed for

this purpose. The match function is based on the mean and standard deviation of a number of pixels on either side of an hypothesised line, and on the direction of the hypothesised line. We show that this match function gives a good variation of match values, producing high valued ridges on a low valued background.

7.2 Preliminaries

To establish any form of match between two images we first need to determine two things:

1. Some features (such as points, lines or regions) to match. These features could be all the points in the image, but generally some particular features are selected.

2. A set of attributes for each feature. These attributes may include, for example, the grey value of the point, position, gradient (magnitude and direction), and texture measures such as the mean and standard deviation of the grey values in a region.

The features will vary depending on the application. For most applications features are taken to be line segments or boundaries between two distinct regions, such as the object and the background; others use points which characterise corners or points of high curvature [93]. For other applications, such as matching cells on consecutive sections, useful features would be the centres of mass of each cell [105, 106]. The measurements used for matching will also vary depending on the application. If the match points characterise edge segments then the features used for matching generally include the edge direction; for matching cells, the radius of the cell, or the relative positions of the cells may constitute useful measurements.

In chapter 3 we saw that there were two main approaches to matching: an approach based on matching features, such as lines, points or regions, to similar features in the other image, and an approach based on matching points with associated attributes to a region in the second image. Matching grid points using cross correlation [99, 19] is an example of the latter case. Due to difficulties in reliably segmenting biological images so that the same features are represented in both images, we will consider the second

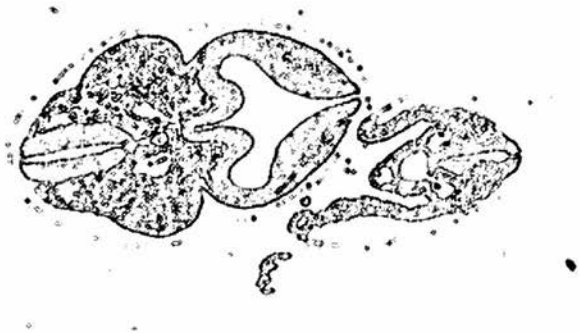


Figure 7.1: Gradient image smoothed by a Gaussian of half-width at half-maximum 1.0.

approach.

There is little point in matching all the points in image 1 to image 2 because points in regions of background or homogeneous texture should match equally well to a number of points in the corresponding region, giving (if the images are registered) little information regarding the movement required to bring such a point into alignment. For this reason we will aim to match only those points which will match either to a specific point or to an edge. For our application we require that the features should be easy to extract from a wide range of images and so we will choose to match points of high gradient which should ideally lie on boundaries between regions. However, as can be seen from figure 7.1, most of the points in the object have a high gradient. This makes it very difficult to ensure that all the points on the boundaries between regions are found and that no points in regions of texture are found. Fully automatic contour extraction is a very difficult problem for biological images in general, although there are efficient algorithms for some specific situations [105, 106, 100]. We will not attempt to solve this, but instead we propose an algorithm which will ensure that many boundary points are found, while allowing some texture points. This algorithm will be discussed in section 7.5. We will denote these chosen points as *interesting points.*

The aim of matching is to provide information so that the forces (which we will model as springs) acting on each section can be calculated. To attach springs we need a corresponding point or line and a strength for the spring which should be derived from

the likelihood that the correspondence is correct. We cannot expect the position of the best match to give the correct correspondence, due to deformation of the sections, which may cause points on edges to match best to a different part of the edge as it is deformed. Therefore we will have to analyse the matchmap, that is the image produced when one point is matched to a region of points in the other image, in more detail. If we require that a good match gives a high match value, we would prefer to attach the spring either to the closest point on the nearest ridge, or to the closest peak, depending on the type of interesting point being matched. This involves either identifying the peaks and ridges in the matchmap or summing over the whole matchmap so that an attraction to the closest ridge or peak is calculated. Since it is essential that the correspondence calculation is fast, and to avoid the problems associated with segmentation and detailed analysis of the resulting image, we will take the latter approach. In order to sum over the matchmap it is essential that the places where the point matches well have significantly higher values than those where the point does not match well, otherwise the corresponding point will always be close to the centre of mass of the matchmap. In other words we need matchmaps which consist of high valued ridges or regions on a low valued background so that only points which match well contribute significantly to the calculation.

We do not expect large changes in the direction of the edges in the sections because we are assuming that the sections have been registered. Deformation should not cause large localised changes in edge direction as this would produce a large torque over the section. Therefore the penalty function with respect to differences in gradient direction should be minimum when the directions are equal.

Since it is very difficult to ensure that there are no interesting points in homogeneous textured regions (that is regions where there is no obvious structure to the texture), we must consider the type of matchmap required when these points are matched. In this case we require a matchmap which will enable such points to be given a verdict of "no match" signifying that they do not match to any particular point. We would expect points in regions of homogeneous texture to match badly to some neighbouring points but well to others producing a matchmap with a scatter of high values. The same should be true of background values. We also need a verdict of "no match" for points which are interesting but which do not match well to anything and should therefore

give a relatively flat matchmap. We will discuss how a verdict of "no match" can be obtained from such matchmaps in chapter 8.

The requirements of the matchmap therefore are:

- The matchmap should ideally consist of steep high valued ridges and peaks on a low valued background.
- There should be a low penalty for small differences in orientation so that corresponding points can match anywhere along a line.[1]
- Points in homogeneous regions such as the background or regions of texture should match either equally well to all points in their corresponding regions, or alternatively there should be a scatter of points where they match well. Similarly for points which don't match to anything.

Some work has been done on matching similar biological images. Most of these methods rely on either user provided tie points [26] or on image matching functions based on cross-correlation [5, 70, 103]. Although these matching algorithms are designed to match similar (biological) images, cross-correlation is not suited for matching distorted images because it is sensitive to minor changes in the images. Cross-correlation also gives a flat peak causing features to match well to large regions of the image. This will make it very difficult to give a verdict of "no match" for areas of texture, which are not expected to have corresponding points.

Some work, specifically aimed at textured images, has been done on using fractals [25, 56] to segment images based on the mean and standard deviation of greylevels. These methods were not tried because they require large regions of similar texture, and will not work on small regions.

A lot of traditional matching algorithms are based on edge segments found by locating points of high gradient ([16, 3, 63]). These methods work well for images of man-made objects with smooth surfaces. In these types of images, the image gradients are

[1] Corresponding points are allowed to match anywhere along a line rather than being restricted to a portion of the line, because this is easier to include in the finite element method. In practise points will only slide a short distance along the line because of the constraints posed by other matches.

low everywhere except where there is a discontinuity between objects. This is not the case for our biological images where, as we saw above, points of high gradient are also found in areas of texture. Nevertheless, since edges can be seen in the gradient image (figure 7.1), and because the gradients vary less along boundaries than in textured regions, some experiments were performed with image gradients. These experiments will be discussed in the next section.

7.3 Gradient Matching

7.3.1 The Match Function

There are two pieces of information which are associated with each interesting point chosen from the gradient image: the position, and the gradient vector at that point. Two points in adjacent images match well if their gradient vectors have similar direction and magnitude. Since, as discussed above, it is important that small differences in edge direction only give negligible match penalties, the benefit for similarity in direction should be cyclic and give low values for small differences in direction. A simple function which satisfies these requirements is $\cos(\theta_1 \quad \theta_2)$, where θ_1 and θ_2 are the angles made by the two gradient vectors with the x axis. The reason for choosing this measure is that when the difference in the angles is close to zero, the value of the cosine function is close to 1.0. Indeed the value of this function when the difference in angle is as large as 10 degrees is still 0.98. The benefit for similarity in magnitude was set to $\min\left(\frac{|\mathbf{v}_1|}{|\mathbf{v}_2|}, \frac{|\mathbf{v}_2|}{|\mathbf{v}_1|}\right)$, where $\mathbf{v}_1$ and $\mathbf{v}_2$ are the two gradient vectors. This gives 1.0 when the vectors have the same length and a low value if their lengths are very different. The actual match function employed was the product of the two functions described above:

$$m_1 = \cos(\theta_1 - \theta_2) \times \min\left(\frac{|\mathbf{v}_1|}{|\mathbf{v}_2|}, \frac{|\mathbf{v}_2|}{|\mathbf{v}_1|}\right).$$

This gives values between -1 and +1 where values less than zero signify that the two points don't match and values close to 1 signify that the two points match well.

The above formula gives a measure when two points are matched. Now when matching one image to another, we would expect that matching a neighbourhood of points in one image to a neighbourhood of points in the other image to work better because matching neighbourhoods should smooth out any spurious gradient values. For this

reason matching neighbourhoods of gradient vectors was tried and the formula tested was

$$m_2 = \frac{\sum_i \cos(\theta_{1i} - \theta_{2i}) \times \min\left(\frac{|\mathbf{v}_{1i}|}{|\mathbf{v}_{2i}|}, \frac{|\mathbf{v}_{2i}|}{|\mathbf{v}_{1i}|}\right) \times \min\left(|\mathbf{v}_{1i}|, |\mathbf{v}_{2i}|\right)}{\sum_i \min\left(|\mathbf{v}_{1i}|, |\mathbf{v}_{2i}|\right)},$$

where the summation is over all the points in each neighbourhood (both neighbourhoods should be the same size), and where the match is weighted by the lengths of the smallest vectors giving more weight to larger gradients. This should ensure that the edge gradients in the neighbourhood are given more weight than background vectors which should be smaller. Note that this measure is similar to correlation except that it is defined over vectors.

7.3.2 Experiments

Nine points were selected manually as examples of the types of situation the match algorithm would have to deal with. These points and the 25×25 regions over which they were matched in the second image, are depicted in figure 7.2. They are:

1. an external edge point;
2. a point which should match in the second image, but doesn't due to a large artefact in that image;
3. an internal edge point;
4. an edge point which should match well to more than one nearby edge in the second image;
5. a point on the membrane (an example of a thin greylevel ridge);
6. an external edge point near part of the membrane;
7. a point in the background;
8. a point in an area of texture;
9. a point on an artefact which corresponds to nothing in the second image.

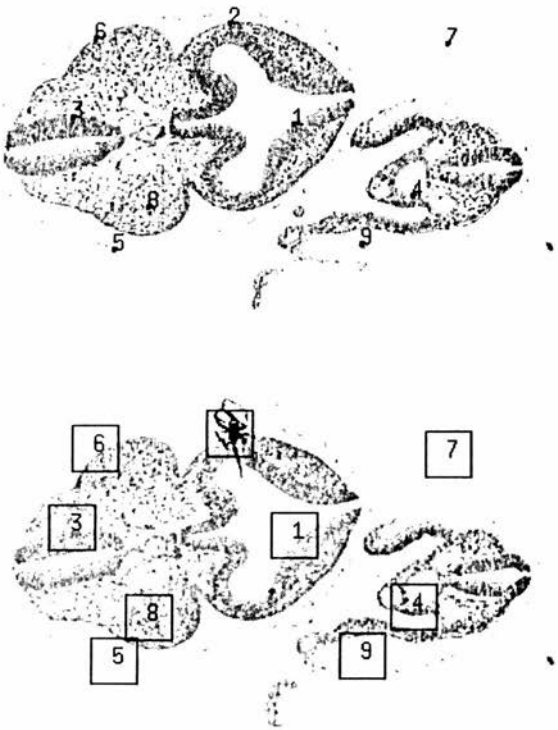


Figure 7.2: The top image shows the points which will be matched to the regions shown in the bottom image.

Two sets of experiments were performed with the two match functions m_1 (experiment 1) and m_2 (experiment 2). In both experiments, the images were filtered with a derivative of Gaussian filter of half-width at half-maximum (HWHM) of either 1.0 or 3.0 pixels. In the second set of experiments the summation in m_2 was over 5×5 neighbourhoods. (This required matching over 29×29 pixel regions in the second image.)

7.3.3 Results

Although negative values give important information (i.e. the point does not match), in view of attaching a spring to a peak or ridge, we are more interested in the characteristics of the peaks. Therefore in order to make the peaks stand out more clearly, the negative values in the matchmaps have been set to zero in all cases. For display purposes, all the positive match values were multiplied by 254.

Experiment 1

The results of matching the nine points are given in figures 7.3 and 7.4 where the HWHM of the Gaussians were 1.0 and 3.0 respectively. We see that the smoothed version gives a better matchmap for our purposes in that the ridges stand out more clearly from the other matches, and point 3 (an edge point within the tissue) has a clear matching ridge. However we also see that the ridges are much wider and that points 2 and 9 have more convincing matches, whereas they should not match at all.

Experiment 2

The results of matching the neighbourhoods of the nine points are given in figures 7.5 and 7.6. Again, we see that the use of smoothing causes the ridges to stand out more while also blurring them. However in this case, the values of the ridges denoting the correct corresponding edges are less than those obtained when matching single points. In contrast the values of the ridges denoting false matches have not decreased and in some cases have increased in value, thus increasing the competition from false matches. We conclude that these matchmaps are less suitable for our purposes than those obtained by matching single points.

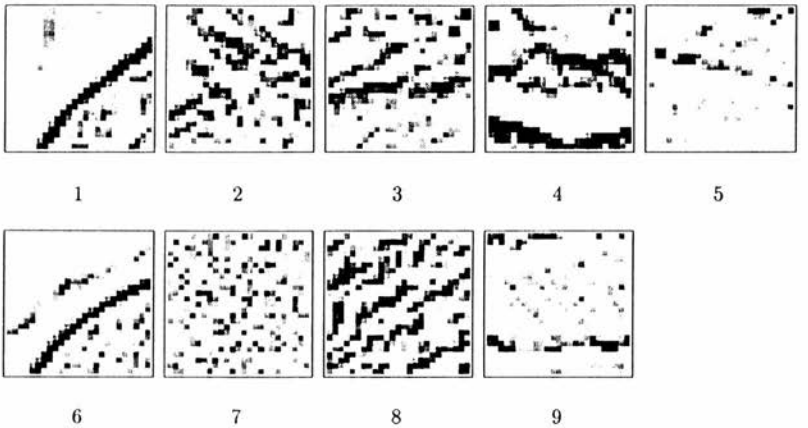


Figure 7.3: Matchmap produced by using m_1 to match the points shown in figure 7.2. The images were smoothed by a Gaussian of HWHM 1.0 prior to matching.

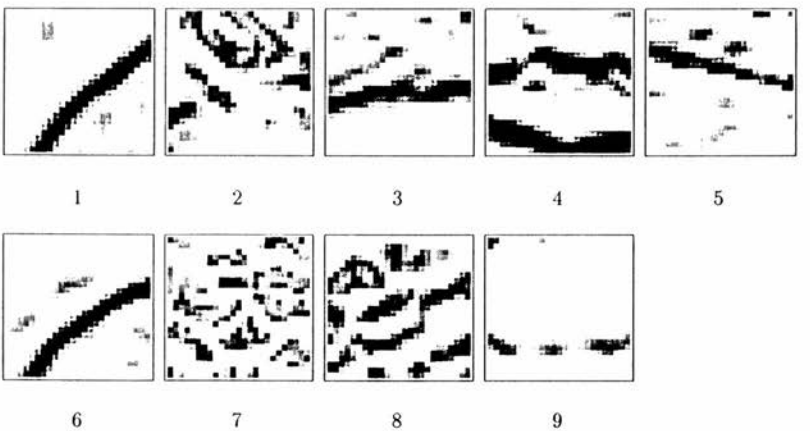


Figure 7.4: Matchmap produced by using m_1 to match the points shown in figure 7.2. The images were smoothed by a Gaussian of HWHM 3.0 prior to matching.

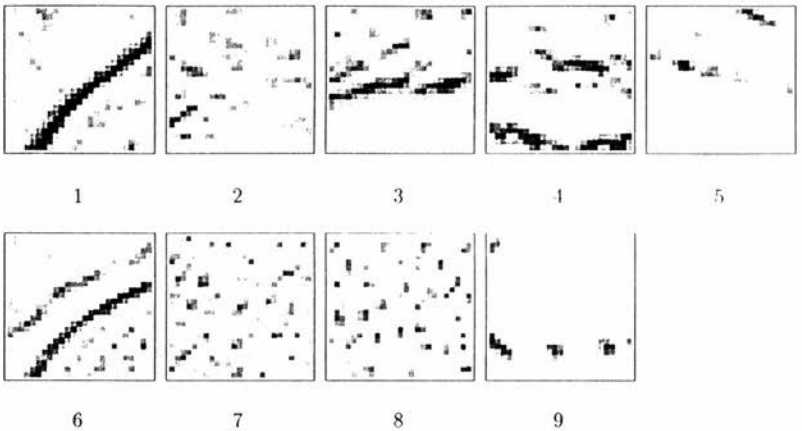


Figure 7.5: Matchmap produced by using m_2 to match the points shown in figure 7.2 using a 5×5 neighbourhood. The images were smoothed by a Gaussian of HWHM 1.0 prior to matching.

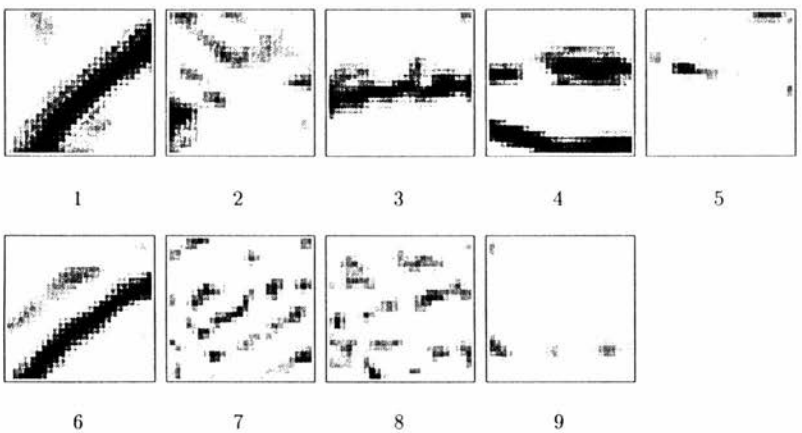


Figure 7.6: Matchmap produced by using m_2 to match the points shown in figure 7.2 using a 5×5 neighbourhood. The images were smoothed by a Gaussian of HWHM 3.0 prior to matching.

7.3.4 Discussion

From the point of view of calculating correspondences these matchmaps do not meet our requirements. The problem with the matchmaps resulting from smoothing the images by a Gaussian of HWHM 1.0 is that the ridges do not stand out well from the background, and as a consequence any attraction of the interesting point to a corresponding line will be adversely influenced by false matches. Although additional smoothing causes the ridges to stand out more, the resulting matchmaps are unsatisfactory for two reasons: the ridges become wider causing some false matches to become amalgamated with the ridge, and information contributing to the verdict of "no match" is deleted causing false matches to dominate the matchmap (as is the case for points 2 and 9). The first problem can be alleviated by filtering the matchmaps so that only the skeleton of the ridges remain. But the result of such a filter would cause problems when dealing with background and texture matchmaps as further information contributing to the verdict of "no match" will be deleted. We note that the results for the less smoothed images gave better matchmaps from the point of view of achieving a verdict of "no match". Clearly, the degree of smoothing could be optimised to enhance the ridges while minimising the loss of information contributing to verdicts of "no match". However it is likely that such a parameter would have to be finely tuned for each set of images, and we would prefer a more robust matching algorithm.

The reason why these results are not as good as we would like is that as can be seen from figure 7.7, which shows the x and y gradients for an image, the gradients within the textured parts of the object are of variable magnitude and direction. This means that points between two regions internal to the object will match well to points in both regions, but possibly not so well to points along the internal boundary due to variations in the gradients along that boundary. This is why the corresponding line for point 3 was only found when the images were smoothed. However smoothing the images is not the solution as this causes some of the internal edges to disappear, while regions of texture seem to become more structured as the finer details are smoothed out. This can be seen in figure 7.8 which shows an image smoothed by a Gaussian of HWHM 3.0.

For the reasons outlined above gradient matching was found to be unsuitable and an

Figure 7.7: The top image shows the x derivatives; the bottom image shows the y derivatives. In both case blue signifies a negative value and red signifies a positive value. The derivatives were calculated using a derivative of Gaussian filter of HWHM 1.0.

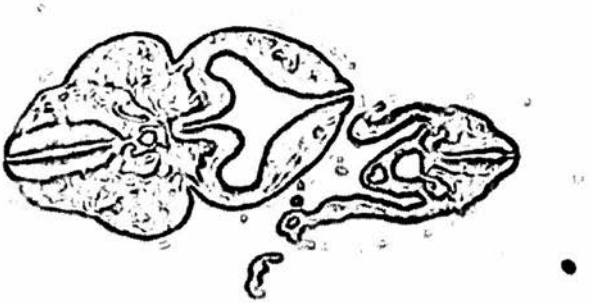


Figure 7.8: Gradient image smoothed by a Gaussian of HWHM 3.0.

alternative method was sought.

7.4 A New Matching Algorithm

In section 7.3 we saw that gradient matching was unsatisfactory because edge points matched to points in areas of texture as well as to other edge points. In order to produce a satisfactory matchmap we need to prevent edge points from matching well to other points. A solution to this problem is described below, and involves characterising the greylevels of different parts of the object using the texture measures, mean and variance, on either side of an hypothesised line. Matching involves comparing these measures.

7.4.1 Feature Measurements

From the images in figure 7.2 we see that the pixels in the different regions of the object have different texture. One way of characterising the texture of each region is by the mean and variance of the greylevels within each region. Other texture measures could have been employed, but the mean and variance proved to be sufficient. In order to find a boundary between two regions, we need to be able to calculate the likelihood that two subregions are part of the same parent region by comparing their texture. Given such a measure, we can locate a boundary by calculating the likelihood that that two regions on either side of an hypothesised line are from different regions, and by calculating this likelihood for a series of edge directions through a point, we may obtain

a measure of confidence that the point is part of a boundary and an estimate of the boundary direction at that point. Interesting points can then be selected by choosing points which are most likely to lie on a boundary and the measurements available for use with a matching algorithm, which would compare the regions on both sides of the hypothesised edges, include the mean and variance of each region. Note that with the exception of the hypothesised line direction all the measurements are invariant to rotation and scaling.

We now consider how this idea can be applied to discrete images. In order to locate small scale edges, we need to be able to compare small neighbourhoods of pixels, and therefore we employ tests from the statistical theory of small samples. Two tests from this theory which are suitable for this application are the F-test and the T-test [27]. The F-test is defined as

$$F = \max\left(\frac{\sigma_1}{\sigma_2}, \frac{\sigma_2}{\sigma_1}\right),$$

where σ_1 and σ_2 are the variances of the greylevels of regions 1 and 2 respectively. The (2-tailed) T-test is defined by

$$T = \frac{|m_1 - m_2|}{\sqrt{\frac{\alpha(n_1 + n_2)}{n_1 n_2}}},$$

where m_1 and n_1 are the mean and number of pixels in region 1; m_2 and n_2 are the corresponding variables for region 2; and

$$\alpha = \frac{n_1\sigma_1 + n_2\sigma_2}{n_1 + n_2 - 2}.$$

When $n_1 = n_2 (= n)$, this reduces to

$$T = \frac{|m_1 - m_2|}{\sqrt{(\sigma_1 + \sigma_2)/(n-1)}}.$$

By means of statistical tables, these tests can be used to calculate a confidence that two samples came from the same parent region. However both tests give low values if the two samples may have come from the same parent region, and high values if the samples differ significantly, and for our purposes, it is not necessary to convert the values to the statistical probability values (although this may be desirable if edge tracking is required). The T-test compares the means of two samples, but is designed only for samples with the same parent variance; the F-test compares the variances

and has no underlying assumptions about the means or variances. In our application both the means and the variances of the greylevels in different regions are likely to be different. To compare two regions we may use either the F-test, or the T-test[2], or a combination of both. We have chosen to use the F-test. However experiments showed that there was little difference between the results of the two tests, probably because of the small number (typically 10) of pixels in a sample. Since the use of a single test gave satisfactory results, a combination of tests was not tried.

In order to locate small scale edges, we must consider the pixels within a small neighbourhood to determine whether a pixel lies on an edge. However only a small number of independent edge directions can be represented by a small discrete neighbourhood (4 for a 3×3 neighbourhood; 24 for a 5×5 neighbourhood; and 42 for a 7×7 neighbourhood, for example). Therefore to calculate a possible edge direction, we need only pass a series of masks over the image. Examples of masks for a 5×5 neighbourhood are:

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 \\ -1 & -1 & -1 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}, \text{ and } \begin{bmatrix} -1 & -1 & -1 & 0 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & 0 & 1 & 1 & 1 \end{bmatrix},$$

which represent the edge direction angles -45, 30 and 75 degrees respectively. Here the pixels denoted by -1 form one region, those by 1 the other region, and pixels denoted by 0 are ignored. Since the variances for the two regions are calculated separately for each region, we are able to halve the number of masks by using the same mask for two edge directions. As each mask is passed over the image, the F-statistic is calculated and if it is the maximum for that point so far the value is stored. The end result of this filter, which we will call the *F-filter* is an image containing the maximum value for the F-statistic at each point.

In actual fact a variation of the F-test was used. To avoid division by zero, the minimum variance was set to 1.0. This was done by taking the maximum of each variance and 1.0 with the aim of avoiding the detection of low contrast edges which would almost definitely be spurious: the variance in all regions (including the background) is greater

[2] There is a T-test designed for samples with unequal underlying variance. This was tried but the results were similar to the other two tests.

than 1. This restriction causes the minimum value of the F-test to be 1.0, and therefore 1 was subtracted from each F statistic so that a result of 0.0 would signify that the two regions were almost certainly from the same parent region.

The application of the F filter described above raises the question of how to deal with the pixels on the edges of the images. Since the object of interest will be approximately in the centre of the image, and because of the registration error obtained while the images were digitised, we will ignore edge effects by simply calculating the values for the mask over the area where the mask fully intersects the image.

Experiments and Results

Experiments were performed to evaluate the results of applying the F filter with masks of different sizes to the images. The results of applying a 5×5 and a 7×7 mask to an image are shown in figure 7.9. We see that the larger mask has a smoothing effect on the image and that the outer boundaries show up more strongly. The smaller mask was chosen for use when matching images because some of the internal edges are more visible, and because it is much faster to compute. Note that although points of high curvature would be useful, they are not found using this method. Corner detectors do exist ([28], for example), but since a separate match algorithm would be required to deal with these points, this possibility was not pursued.

7.4.2 The Match Function

We now consider how the results of the F filter can be used to match two images. For matching purposes, the information stored as a result of applying the F filter to an image may include the means and variances of the regions on either side of the hypothesised lines. This will enable us to base our matching algorithm on the similarities between possible corresponding regions rather than simply on the similarities between two gradient vectors. Therefore given two images, A and B, both filtered with the F-filter, we match any two points, one from each image, as follows. Let regions R_{A_1} and R_{A_2} be the regions to the left and right, respectively, of the hypothesised edge for the point in image A; and let R_{B_1} and R_{B_2} be similarly defined for the point in image

Figure 7.9: An image filtered with the F-filter. The top image was filtered with a 5×5 mask, the bottom image was filtered with a 7×7 mask.

B. The match function described below, measures the similarity between regions R_{A_1} and R_{B_1} and between regions R_{A_2} and R_{B_2}. We require that the match function gives a high value if the two pairs of corresponding regions have similar means and variances and the edge directions are similar.

To calculate a measure of similarity between each pair of regions, we multiply the results of the F-test and the T-test. This allows differences in both the means and the variances to be taken into account and yields low values if the regions may have come from the same parent region and high values of the regions differ significantly. To check that the edge directions are similar, we can use the exponential function of the difference of the angles. This will give unity if the angles are equal (or within the angle limits imposed by the masks) and a very large number if they are very different. Therefore we define the match function to be[3]

$$m = (1 + \text{T-test1}^{\alpha} \times \text{F-test1}^{\beta} + \text{T-test2}^{\alpha} \times \text{F-test2}^{\beta}) \exp\left(\frac{1}{10}\left(\frac{|\theta_1 - \theta_2|}{10}\right)^{b}\right)$$

where T-test1 and F-test1 refer to the results of the T-test and F-test applied to R_{A_1} and R_{B_1}; T-test2 and F-test2 refer to the results of these tests applied to R_{A_2} and R_{B_2}; θ_1 and θ_2 refer to the edge directions; and α, β, and b are constants. This function will give 1 for a perfect match and a very large number for two sets of features which don't match. If we now set our match function to

$$M = 1/m$$

we get a function which gives 1 for a perfect match and 0 for no match.

The reason for choosing an exponential function rather than the cosine function for the penalty in the difference in angles, is that the exponential ranges from 1 to very high values as the difference in angle increases. This ensures that if the means and variances of the pixel values are similar but the difference in angle is large, the match value will be low. The constants within the angle penalty part of the formula are designed to cause only a small penalty for differences in angle up to 15 degrees (the value of the exponential is then 1.25 (for $b = 2$)) as that is the angle resolution for masks of size 5×5.

[3] The unequal variance T-test was also tried, but the normal T-test gave slightly better results in that the matchmaps with a clear ridge had fewer competing matches.

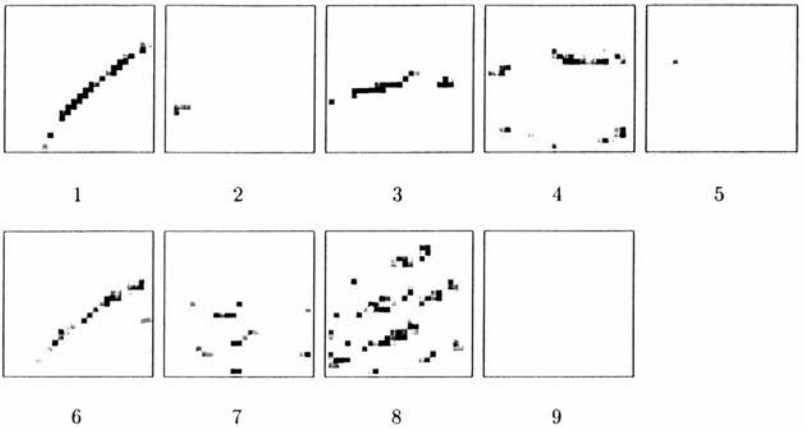


Figure 7.10: Matchmap produced by matching the points shown in figure 7.2 using the new matching algorithm. A 5×5 neighbourhood was used.

Experiments and Results

Experiments to evaluate the effects of different values of the parameters of the match function were performed: α and β ranged from 1 to 3; b ranged from 0 to 4. By subjectively comparing the resulting matchmaps, the parameters were fixed at $\alpha = 1$, $\beta = 2$, and $b = 2$. However it should be noted that, except when the angle penalty was eliminated ($b = 0$), changing the parameters had little effect on the resulting matchmaps. The matchmaps for this new matching algorithm, for the same points as those used for gradient matching, are given in figure 7.10. We see that points 1, 4 and 6 which are points on strong edges produce matchmaps with ridges on a low valued background as required. Point 3 matches to some of the points in the textured region, but the match to the line is stronger. Matchmaps consisting of a scatter of high and low values are produced by the points lying in the background and in a textured region, and the matchmap for point 9, which should not match well to anything in the second image, shows that this is indeed the case. We see that point 2 only matches well to the portion of the corresponding edge contained in the search area; in contrast to the results of matching gradients, it does not match well to the large artefact. Point 5 which is part of the membrane matches only weakly to the membrane in the other image, but

there are few other places where this point matches. From this we conclude that this new matching algorithm produces the required type of matchmap.

Discussion

These matchmaps satisfy our requirements: the ridges are thin, allowing precise location of matching edges, and points in areas of homogeneous texture give matchmaps with a scatter of both good and bad matches. In addition, when a point matches well to an edge it does not match well to other points, and in particular we note that, in contrast to the results of gradient matching, point 2 does not match well to the artefact in the other image.

These matchmaps clearly meet our requirements better than those obtained using gradient matching (although the new method uses a type of gradient). The reasons for this lie in the different methods of calculating edge strengths, and in the additional information of the distribution of greylevels on either side of an hypothesised line provided by the new method. By calculating edge direction and magnitude from analysis of the greylevels of groups of pixels, we are able to ignore small scale changes without compromising on the precision of edge location. This effect should also cause the matching method to decay gracefully as the amount of noise increases. This is indicated by the fact that a satisfactory matchmap was obtained for point 3. The additional information of the greylevel statistics of regions either side of an hypothesised line allows the number of false matches to be reduced because matching these greylevel regions ensures not only that a candidate matching edge has the correct direction and edge strength, but also that the regions on either side of the lines are compatible.

Finally we remark that this new method is not sensitive to the values of the parameters, and therefore will not have to be tuned to suit each set of images.

7.5 Finding Interesting Points

We need to find a way of identifying a suitable number of interesting points for matching purposes. As discussed in section 7.2, the interesting points should be located on the

boundaries between regions, and we would prefer few interesting points to be found in the background or other homogeneously textured regions. Clearly matching every point on an edge would be very time consuming, and we would prefer interesting points to be scattered evenly along the edges. As discussed above a satisfactory way of identifying interesting points has not yet been developed. For the purposes of this project we have chosen to identify them simply by picking points with high gradient values subject to not being too close together and subject to not being part of the background. The algorithm is as follows. All thresholding is performed automatically as described in section 5.2.1.

1. Threshold the object to remove background.

2. Calculate the modulus of the gradient of the greylevels (here a derivative of Gaussian of HWHM 1.0 was used).

3. Filter the image using the F-filter described in section 7.4.1, and thin the result so that only the highest points on the ridges remain. The algorithm to thin the filtered image exploits the edge direction information by insisting that the pixels on the ridge must have higher values that the neighbouring pixels in directions perpendicular to the ridge (non-maximal suppression).

4. Intersect the domains (that is the parts of the images with non zero pixel values) of the three images obtained in stages 1 to 3. The remaining points are assumed to belong to the object; all other points are assumed to belong to the background.

5. Order the points obtained in stage 4 by their values in the image obtained in stage 3. Then starting with the point with the highest value, consider each point in turn rejecting those that are within a minimum distance of previously selected points.

The results of applying this point finding algorithm are shown in figure 7.11. We see that the points are scattered evenly along the edges and within the object. There are few points in the background, and those that are, correspond to artefacts.

Although the algorithm described above is adequate for the purposes of this project it should be possible to design a better algorithm for identifying interesting points. One

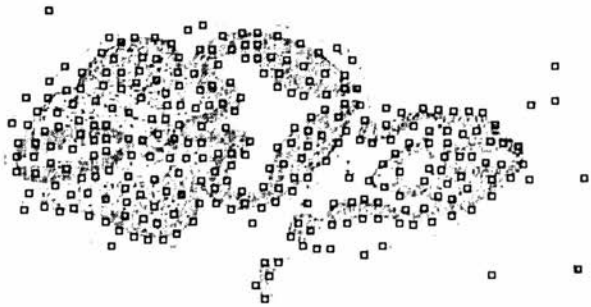


Figure 7.11: The 273 interesting points found for this image.

approach would be to use the measurements associated with each pixel. For example we could insist that the neighbouring pixels (in an appropriate direction) have similar edge directions.

7.6 Conclusion

A new match function for matching distorted biological images has been developed. This match function is designed to match points on boundaries between regions with distinct pixel distributions and is based on the means and standard deviations on either side of an hypothesised line. Matching such points gives a good variation of match values, producing high valued ridges on a low valued background; matching points in the background or in areas of texture produces a matchmap consisting of a scatter of high and low values; and points which should not match produce a flat matchmap. This type of matchmap enables us to calculate correspondences so that springs can be attached between points and their corresponding lines. The details of the calculation of correspondences will be discussed in chapter 8.

Chapter 8

Force Calculation and Iteration.

In chapter 7 we discussed the problems of matching adjacent sections in such a way as to ensure that the resulting matchmap is suitable for calculating correspondences and forces. In this chapter we outline the requirements of the force calculation and describe the problems which must be overcome. A solution to these problems is given, and finally we discuss the problems of iteration.

8.1 Requirements and Characteristics.

We begin by discussing the requirements and the required characteristics of the force calculation. As discussed in chapter 5, we are modelling the forces between the sections as springs. In order to attach a spring we need a corresponding point or line to which to attach the spring, and a spring constant. Since the matching algorithm described in chapter 7 matches edges, we would prefer to attach the spring to a corresponding line. This line has to be infinite in order to simplify the finite element calculations, and since we are dealing with closed biological surfaces, this should not cause any problems. Each spring requires a spring constant. Since it is unlikely that every correspondence will be correct, the spring constant should depend on the likelihood that the correspondences are correct. In the event that there is a choice of corresponding line, we would expect the spring strength to be small until the interesting point becomes sufficiently close to one of the lines that the correspondence becomes clear.

Since it is unlikely that all the initial correspondences will be correct, the correspond-

ences must be allowed to change as the section is warped, and they will have to be updated continually. In order to allow the warping process to converge, we must insist that the force calculation be stable so that small perturbations in the position of the interesting point will not lead to major changes in either the position of the corresponding line, or the value of the spring constant. In particular, for correct correspondences we would expect the spring constant to increase or remain constant as the sections are warped. In this case, the corresponding line should also remain constant.

We have established above that the requirements of the calculation are

- a corresponding line for the point to slide along, and
- a spring constant.

We first consider the correspondence calculation. The aim is to calculate correspondences in order to provide the lines to which springs will be attached. We assume that the sections are registered and that therefore each match lies in a small neighbourhood centered on the point to be matched. In this case, closer lines which match well will be preferred over those which are further away, and therefore these lines should be given a higher weight than more distant points. That is, match strengths *decrease* with distance.

We now consider the problem of using information from both adjacent sections to warp the middle section. The underlying assumption is that surfaces are smooth on the scale of the section thickness, and therefore we expect a set of three matched points, one on each of three consecutive sections, to lie on a straight line if no other influences are present. Clearly to achieve this, correspondences which are more distant should pull more strongly than those which are closer. That is, match strength should *increase* with distance.

Therefore when comparing match possibilities *within* a single adjacent image, the match strength must *decrease* with distance, but when estimating the forces to attach to a point arising from both adjacent images, the match strength must *increase* with distance. To satisfy both these constraints, we model the forces as springs for which a greater extension implies greater force (satisfying the second constraint) and, to satisfy the

first constraint, we estimate the spring constant using the match probability. This probability is calculated using a two stage approach. First a "tentative corresponding point" is calculated for an interesting point by integrating over the matchmap using a formula which gives greater weight to closer matches. Then the match probability is calculated by testing the sensitivity of this tentative corresponding point to movement of the interesting point. This approach allows the first constraint to be satisfied but avoids "double counting" as it does not depend directly on distance.

8.2 Problems with Summing over the Matchmap

In this section we show that it is very difficult to calculate both a corresponding line and a suitable spring constant using a simple summation over the matchmap. In the next section an alternative solution will be presented to these problems.

We have established that we need to calculate two things: a corresponding line and a spring strength, both of which have to be calculated from the matchmap. The matchmap was designed to consist of high valued ridges on a low valued background so that it would be possible to integrate over the matchmap to calculate attractions to possible corresponding lines.

8.2.1 Summing over the Matchmap

Assuming that the interesting point is positioned in the centre of the matchmap, a displacement vector can be calculated by doing a simple integration such as

$$\mathbf{v} = \frac{\sum w_i m_i \mathbf{x}_i}{\sum w_i m_i},$$

where m_i is the match value at position $\mathbf{x}_i$ and w_i is the distance weighting function. The summation is over all the points in the region to which the interesting point is being matched. The distance weighting function should be designed to ensure that the interesting point is attracted to the closest point on the nearest ridge in the matchmap. Without the distance weighting function, the displacement vector would be to the centre of mass of the matchmap. Since the length of the $\mathbf{x}_i$ is included in the calculation, the distance function should decrease faster than the length of the vector increases. This is

easy to see by writing the above equation as

$$\mathbf{v} = \frac{\sum w_i m_i |\mathbf{x}_i| \hat{\mathbf{x}}_i}{\sum w_i m_i},$$

where $\hat{\mathbf{x}}_i$ is the unit vector in the direction $\mathbf{x}_i$. Then if $w_i(\mathbf{x}_i)$ decays at the same rate as $1/|\mathbf{x}_i|$ then $\mathbf{v}$ will be in the direction of the centre of mass of the matchmap; if $w_i(\mathbf{x}_i)$ decays slower than $1/|\mathbf{x}_i|$ then greater weights will be given to more distant points; if $w_i(\mathbf{x}_i)$ decays faster than $1/|\mathbf{x}_i|$ then greater weights will be given to closer points as required. The simplest way of ensuring the correct behaviour is to set

$$w_i(\mathbf{x}_i) = 1/|\mathbf{x}_i|^a,$$

where a is a constant and $a > 1$, so that

$$\mathbf{v} = \frac{\sum \frac{m_i}{|\mathbf{x}_i|^a} \mathbf{x}_i}{\sum \frac{m_i}{|\mathbf{x}_i|^a}}. \tag{8.1}$$

We will call a the *distance penalty parameter*.

There are two ways of interpreting the vector which results from using this formula to sum over the matchmap: we could either interpret it as a displacement vector indicating the corresponding point, or as a force vector with direction and magnitude. The advantage of considering the vector as a displacement vector is that a spring can be attached to the corresponding point and the force exerted by the spring will change during the finite element calculations. If the vector is considered to be a force vector, the force applied to the interesting point will remain constant in magnitude and direction throughout the finite element calculations, necessitating that the forces be carefully tailored to the material properties to prevent over warping. For this reason, in the following discussion, we will consider the vector to be a displacement vector, although most of the discussion will also apply to the case where the vector is considered to be a force vector.

8.2.2 Results

The result of applying formula 8.1 with $a = 2$ to the same nine interesting points used in the previous chapter are given in figure 8.1. We see that for points 1, 3, 5, and 6 the corresponding point is on the nearest ridge of the matchmap, even when, as is the case

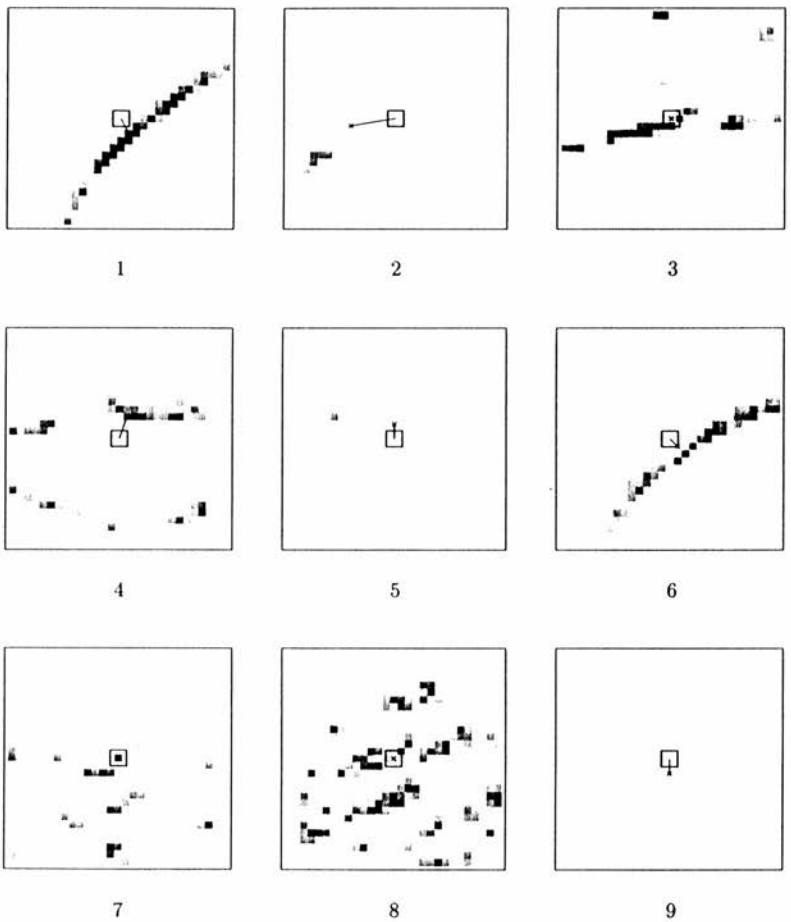


Figure 8.1: Figure showing the corresponding points for the same interesting points used in the previous chapter obtained by integrating over the matchmap. The centre of the small black square denotes the position of the interesting point; the black cross denotes the corresponding point.

for point 6, there is fairly significant competition. The corresponding point for point 4 is located towards the nearest ridge. In this case the pull of the alternative ridge is too strong to allow the corresponding point to be located on the nearest ridge, but this does not matter because if it is correct, it should be closer at the next iteration. The corresponding point for point 2 is towards the corresponding ridge to which it matches. However the real corresponding point is obscured by the artefact in the other image, and here we may prefer a verdict of "no match". In the case of points 7 and 8 which are located in areas of texture, the corresponding points are very close to these points. This is what is required as these points should not be strongly attracted to another position. The corresponding point for point 9 is spurious.

8.2.3 Discussion

Clearly the best value of a will depend on the complexity of the images, the amount of deformation, and the registration of the images. If the images consist of many edges which are close together, the value of a should be higher to reduce the contribution from more distant edges. Conversely if the images are not well registered, or if the images are very deformed, the value of a should be smaller. The best results may be obtained by increasing a as the sections are brought into alignment. This is difficult to implement in practice because there are no good criteria for determining how closely two images are aligned to each other. However for this particular application a suitable measure would be the amount of warping performed during the previous force iteration.

Although the above calculation may provide a reasonable corresponding point in most cases, it is not suitable for determining the forces because it does not provide a line direction or a spring constant. To provide a line direction, the edge direction of the interesting point could be adopted. Calculating spring constants is more problematic. Summing over the matchmap can provide a correspondence point, given by the direction and magnitude of the vector, but in this case it gives no information about how reliable that correspondence point is. This information is essential for calculating spring constants which should depend on the likelihood that the correspondence is correct.

A more serious problem with this approach of summing over the matchmaps is that if the displacement vector is zero or close to zero we cannot tell whether this is because

the point should stay where it is, as in the case of point 3, or whether the interesting point matches equally well (or badly) to all (or a number of scattered) points in the matchmap, and should therefore not be fixed to its current position. The latter scenario is the case for points 7 and 8.

A related problem is that the match value in position $(0, 0)$ does not contribute anything to the calculation, and therefore cannot add weight to the hypothesis that the interesting point should stay where it is. In effect the only way of obtaining a zero direction vector is when the attractions to the surrounding points cancel each other out. As this is unlikely, changing the coordinates of the interesting point may change the correspondence leading to possible problems with oscillation.

The above discussion suggests that it is difficult to calculate both a line direction and a spring constant by simply summing over the matchmap. The main problems are finding a stable correspondence and calculating a spring constant.

8.3 A Solution

In section 8.2 we found that a single calculation does not give a stable correspondence in that small movements in the position of the interesting point can lead to unwanted changes in the correspondence. In this section, we show how this behaviour can be exploited to give both a corresponding line and a spring constant. The basic idea is to test the sensitivity of the correspondence to spatial movement.

8.3.1 Calculating a Corresponding Line

In section 8.2 we saw that when the interesting point lies on an edge, the corresponding point calculated by summing over the matchmap lies on the corresponding edge (as long as there is no choice of matching edge); and if the interesting point lies in a textured region, the position of the corresponding point is close to that of the interesting point. If we were to move the interesting point by a few pixels and recalculate the corresponding point (using an appropriate matchmap), it would, in general, move: the corresponding point for an edge point would lie on a different part of the edge, and for a texture point, it would be close to the new position of the interesting point.

Therefore, if "tentative" corresponding points are calculated for the interesting point as it is displaced by a few pixels in each direction from its original position, the resulting scatter of points will give us information concerning the stability of the correspondence. If the interesting point lies on an edge, the resulting scatter of points should be clustered along a line; if the interesting point lies in an area of texture the points should be widely scattered. In the case where the points are clustered along a line we will be able to calculate a line direction by finding the first principal axis of the scatter of points, and the direction of this line will be reliable because if the interesting point is moved a little, the tentative corresponding points will be scattered around the same line. This line can also be calculated for a widely scattered set of points. In this case, the direction of the line will be unreliable, but this is not a problem because the distribution of the points around the line will enable us to determine its reliability.

To define a corresponding line we need both a corresponding point and a line direction. The corresponding point could be positioned either at the centre of mass of the scatter of points, or at the tentative corresponding point for the match point in its original position. We have chosen to use the centre of mass because it is more likely to remain constant when the correspondence is subsequently recalculated. This is especially important when the interesting point lies in the background or in an area of texture.

Results

The results of this approach can be seen in figure 8.2 which shows the same nine matchmaps as before. The crosses denote the positions of the tentative corresponding points (calculated using equation 8.1) as the interesting point is moved by one or two pixels in each direction (giving 25 different positions). The black line denotes the corresponding line. We see that where there is a single choice of corresponding line (points 1 and 6), the line is correct. The line is also correct for point 2 despite the very limited information due to the artefact in the other image. The line for point 3 is not perfect due to the characteristics of the internal edge. In the cases where there should be a verdict of "no match", as for point 7, 8 and 9, the line direction is arbitrary, but the tentative corresponding points are widely scattered showing that the correspondence is not likely to be correct, which is also the case where there is a choice of corresponding

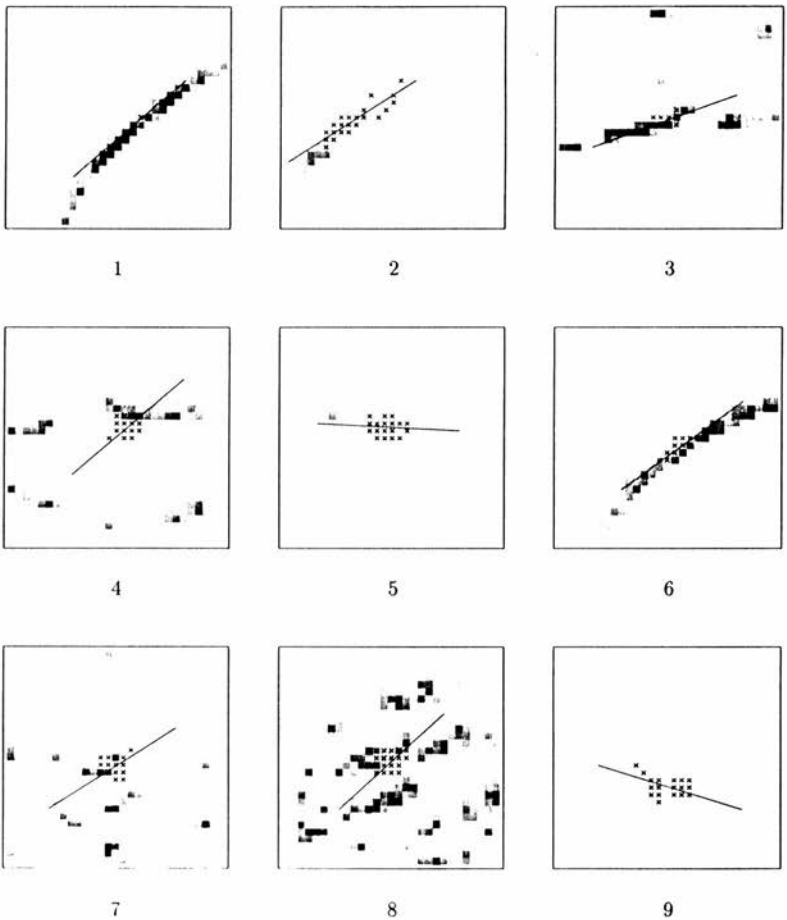


Figure 8.2: Figure showing the tentative corresponding points and the corresponding line for the nine interesting points. The black crosses denote the tentative corresponding points. The black line denotes the corresponding line.

line (point 4). Point 4 is the only case where the result is unsatisfactory. However, the tentative corresponding points are not scattered along the line, throwing doubt on the correspondence. This should enable us to assign a low spring constant to the spring which will be attached to the line so that the interesting point can move relatively freely to enable other correspondences to pull it closer to the correct line. It is in this case that the distance penalty parameter in the summation formula plays such an important role. If the distance penalty parameter is too weak, the interesting point will remain attracted to both lines.

8.3.2 Calculating a Spring Constant

We now consider the calculation of the spring constant. Ideally this should reflect how the tentative corresponding points are scattered around the corresponding line. The spring constants should be in the range 0 to 1, so that the constants can be easily tailored (by multiplying by a constant) to the material properties assigned to the section.

A simple measurement of the spread of a number of points around a line is the sum of the squared distances of the points from the line. This is equivalent to the smallest eigenvalue of the scatter of points. However this measure ranges from 0 to infinity, where 0 indicates that all the tentative corresponding points are located on the line, and higher numbers indicate that the correspondence is less likely. A way of normalising these values is to divide by the largest eigenvalue. This will give 1.0 for a circular scatter of points and 0.0 for a line. Therefore

$$1 - \frac{\lambda_2}{\lambda_1}$$

where λ_1 and λ_2 are the eigenvalues, and $\lambda_1 \geq \lambda_2$, may give a good value for a spring constant. When this was tried, some high values (up to 0.78) were found for some background points. The reason for this is that the above measure effectively normalises the width of the scatter of points by the length of the scatter of points, giving different values for configurations with the same width but different lengths: longer configurations give higher values. For our purposes we require that configurations with the same width give similar values. We note that if both the length and width are small, we can assume that the interesting point matches to a specific point. In this case the model we

are using to model correspondences is inappropriate, but it is a valid correspondence nonetheless. We will not check for this condition since we are matching edges.

Therefore, we will consider other functions to normalise the smallest eigenvalue to the range [0,1], where 0 indicates that the correspondence is very unlikely. There are a number of possible functions which could be used for this task, two of which are the Gaussian function and the reciprocal $(1/x)$ function. By manually selecting interesting points and comparing the smallest eigenvalues of the resulting scatter of points, it was found that interesting points located in the background or in areas of texture generally gave values of 18 and above. Interesting points located on edges generally gave values of less than 5. However in some cases, although the tentative corresponding points were clearly scattered along the line, they were not close to the line giving higher values of around 10 (we see this for point 2 in figure 8.2). As the sum of squared distances function increases quickly when the points are only a short distance from the line, we do not want the conversion function to be too steep for lower sums. Taking this into consideration we conclude that the Gaussian function is a good choice. The HWHM of the Gaussian was chosen to be 10 so that interesting points, which produce a scatter of points which are clearly scattered along a line, but which are not close to the line will be assigned a spring constant of around 0.5, while interesting points in the background and in areas of texture will be assigned spring constants close to zero. Note however that warping a stack using a Gaussian of HWHM 20 was found to give almost the same warp as when a Gaussian of width 10 was used, showing that the results are not sensitive to the value of this parameter. The smallest eigenvalues of the scatter of points and the spring constants for the nine interesting points are given in table 8.1. The values for $1 - \frac{\lambda_2}{\lambda_1}$ are also given for comparison.

8.3.3 Summary

In summary the force calculation algorithm is

1. Integrate over the matchmap and calculate a tentative corresponding point using equation 8.1.

2. Repeat (1) for various different positions of the interesting point.

point	eigenvalue	spring constant	$1 - \frac{\lambda_2}{\lambda_1}$
1	0.08	1.00	0.998
2	10.39	0.47	0.969
3	2.35	0.96	0.940
4	22.86	0.03	0.288
5	14.24	0.25	0.762
6	5.09	0.84	0.888
7	18.32	0.10	0.464
8	31.50	0.00	0.269
9	20.98	0.05	0.758

Table 8.1: Table showing the smallest eigenvalues and spring constants for the nine interesting points.

3. Find the centre of mass and the main principal direction of the resulting scatter of points. This gives a corresponding line.

4. The spring strength is inversely proportional to the Gaussian of the sum of the squared distances from the corresponding line (λ_2). The spring strength may be multiplied by a constant to make it compatible with the material properties assigned to the sections.

Note that if identical images are taken, the corresponding line may not pass through the position of the interesting point, especially if it lies in the background or in an area of texture. This is not a problem because if the interesting point lies on an edge, the corresponding line will generally lie within a pixel, and because the correspondences for interesting points in texture regions are given low weight. Therefore a stack of identical images should not be warped significantly, and this was found to be the case (the maximum displacement was about 0.2 pixels).

8.4 Iteration

There are three types of iteration in the warping process: As we saw in chapter 6, iteration is required to calculate the equilibrium position of each section given the applied forces (modelled as springs). We will denote this the *FE iteration*. The forces change as the section is warped, and therefore have to be recalculated until the forces

applied by the adjacent sections are in equilibrium with the stresses within the section. This is the *force iteration*. The third type of iteration is that required to warp the stack as a whole. The final warp of each section depends on the warp of the adjacent sections. Warping all the sections simultaneously is computationally infeasible due to the size of the problem, and therefore iteration must be employed. We will call this the *stack iteration*.

The FE iteration and force iteration were discussed in chapter 6. We note, however, that the image does not have to be recalculated each time the forces are calculated. This is because the features calculated for each point will only change significantly when there are significant changes to the image. We therefore only recalculate the image when the maximum displacement exceeds a certain threshold. A better measure would be the derivative of the displacements because the features do not have to be recalculated when the section has undergone a rigid transformation. However this requires significantly more computation, and since we are assuming that the sections are registered, and that therefore little, or no, further rigid transformation is required, we may assume that the displacements mostly stem from the deformation part of the warp transformation. Since the features are based on the pixel values on either side of an hypothesised line, significant changes in the features will only occur when the line direction changes. The line direction will only change under a significant rotation (since the number of possible line directions equals twice the number of masks passed over the image) and since groups of pixels cannot rotate significantly without other parts of the image being affected, a fairly large threshold can be set. A threshold of 5 pixels was arbitrarily set. However all the images are updated before each new stack iteration.

There are several possible schemes for warping the whole stack:

1. Split the stack iteration into two passes: the first warping all the odd numbered sections; the second all the even numbered sections. For each pass use information from the previous iteration.
2. Warp each section consecutively using the most recent information.
3. Same as (2) but alternate the direction of warping through the stack.
4. Same as (1) except that the sections are warped using information from the last

pass through the stack, rather than the last iteration.

Clearly (3) is a better scheme than (2) because changes at the bottom of the stack can propagate quicker to the top of the stack; scheme (4) is clearly better than scheme (1) because it decreases the amount of work to be done before the warping converges. Since it is not immediately obvious which of (3) and (4) is the better scheme both were implemented and tested on stacks of identical images. For the details of these experiments and the results see chapter 10.

So far we have only attached each section to its adjacent section, and have not constrained the rigid body movement of the stack in any way. Therefore constraints are needed to fix the stack of sections in space. In addition, we need to constrain the top and bottom sections so that they are not warped so that they match their single adjacent section exactly, in so far as the internal stresses will allow. We have chosen to constrain the warp by fixing the top and bottom sections. For the best results this clearly requires that the top and bottom sections are appropriately registered.

8.5 Conclusion

In this, and the last few, chapters an algorithm for warping serial sections has been presented. The basic idea is to model each section as a thin elastic plate, and to apply forces, modelled as springs, to each section. The finite element method is used to model sections as thin elastic plates, and to calculate the equilibrium position of the plate given the applied forces. In chapter 7 we discussed the problems of matching adjacent sections and we presented a new matching algorithm. In this chapter we have shown how the resulting matchmap can be analysed to provide corresponding lines and spring constants, and finally we discussed strategies for warping the stack as a whole.

In chapter 10 experiments designed to analyse the performance of the warping algorithm will be presented, but first we discuss criteria for evaluating these experiments.

Chapter 9

Evaluation Criteria for Experiments

In chapters 5 to 8 we have described an algorithm for warping serial section images. We now need to test the algorithm to see how well it performs and to see how varying the parameters affects the results. However before performing any experiments we need to determine how the results will be evaluated. In this chapter we discuss the types of experiment which will be used to test the warping algorithm, and consider criteria for evaluating the results of these experiments. Since, as we saw in chapter 4, correlation is not ideal for measuring the quality of a reconstruction, a new measure is developed. Experiments are performed in this chapter to test the evaluation criteria; details of experiments for testing the warping algorithm will be discussed in chapter 10.

9.1 Evaluation Criteria

Before discussing the evaluation criteria, we must first determine the types of experiment we wish to perform. Since each section affects, and is affected by, the warps of adjacent sections it is clear that to examine the performance of the warping algorithm, it is necessary to test the algorithm on stacks of sections. However it is also possible to evaluate the effects of changing some of the parameters, such as the resolution of the finite element mesh, by using test images consisting of a (manually) warped image sandwiched between two copies of the original image. In doing this we verify that the

warping process is capable of warping images back to their original versions[1]. Although the algorithm has been specifically designed so that images are not warped so that they match another image exactly, it should be able to do this in specific circumstances such as when the two adjacent sections are identical. It is the fact that the two adjacent images are not identical which should ensure that the middle image is warped so that it is not identical to either image, but is aligned to both.

To warp an image so that it matches another exactly it is important that the balance between the material properties and the spring constants is such that the elastic properties do not limit the magnitude of the warp causing it to fall short of the ideal, that there are sufficient corresponding points, and that the resolution of the finite element mesh is sufficiently small to enable an appropriate non-linear warp to be calculated. Therefore the effects of these parameters can be evaluated using test images.

To evaluate the results of experiments which involve test images, we simply analyse the residual displacement (RD) after the calculated inverse warp has been applied to the original warp. Since we are not interested in how the background has been warped the images were automatically thresholded and the RD was calculated over the domain of the thresholded object. For this purpose the displacements resulting from a warp are obtained by calculating the (float) displacements of each (integer) point in the domain of the original thresholded object. The actual measures used for analysing the RD are the mean and standard deviation. Note that the measures of the mean and standard deviation of displacements are also useful for comparing the differences between two warps.

Objective evaluation of the results of warping a stack of images is difficult. The methods used in the literature for comparing registration algorithms [86, 99] include

- visual evaluation of a cross section through the stack,
- visual evaluation of adjacent sections by superposition,
- cross-correlation, and

[1] We note here that because the stresses within a warped section must be in equilibrium with the applied forces, it is not possible to warp one section so that it becomes absolutely identical to another section. However with suitable choices for the spring constants and material properties, it should be possible to get arbitrarily close.

- analysis of the positions of tie points.

Visual evaluation is subjective and a significant number of sections have to be warped before a visual analysis of the cross section has any value (so that a reasonable image is produced). In addition, if only small differences are visible it can be difficult to discriminate between the results. Superpositioning adjacent sections can give a good indication of how well two particular sections have been brought into alignment, but no indication of how well the whole stack has been reconstructed. The only objective evaluation criteria used in the literature are the mean and standard deviation of the image correlation of adjacent sections, and the absolute differences from the mean position of a number of tie points.

We cannot use tie points because of the time involved in manually marking each tie point and the impossibility of locating them precisely. In chapter 4 we discussed the suitability of correlation for measuring the quality of the alignment in registered stacks of images, and came to the conclusion that it is not ideal. For the same reasons as given in chapter 4 (the problems of matching concentric rings, for example), it is not an ideal measure for measuring the quality of warped stacks of images either. In addition, when warping the images we do not require that the correlation value approaches 1.0 as this will indicate that the images have become identical, which is not the required end point. However up to a point increasing correlation values will indicate better alignment. In particular we would expect the standard deviation of the correlation values to decrease as the stack becomes aligned, because in an aligned stack each section should match equally well to both its adjacent sections (on average).

In chapter 4, we suggested that a better measure for reconstructed images would be one based on the distances between corresponding points. However this only calculates a measure for the alignment of two consecutive images. To gauge how well an image has been aligned simultaneously to both its adjacent images, we need a calculation based on at least three images. In fact, the ideal measure would measure the smoothness in the z direction of the reconstructed surfaces. A fully automatic implementation of this would be difficult because of the difficulties of reliably segmenting surfaces from 3D data. The alternative interactive method of generating surfaces from interactively generated contours was not considered because of the time involved in generating the contours.

Therefore an approximation to this approach was devised. We approximate the smooth surface constraint by assuming that if a point is perfectly aligned, it lies midway between its corresponding points on adjacent sections. This is a reasonable approximation as long as the curvature, in the z direction, of the surfaces changes slowly with respect to the distance between sections, and in particular does not change sign. Since most embryological surfaces meet these constraints most of the time, a measure based on the sum of the displacement vectors from a point to its two corresponding points on adjacent sections should give a reasonable estimate of the quality of a reconstruction.

The new measure based on this approach is as follows: For all pixels in the image which are candidate "interesting points" as described in chapter 7 but without the minimum distance constraint, calculate the corresponding lines and associated spring constants in both adjacent images. If the spring constants (which are related to the match confidence) for both corresponding lines are greater than a threshold, calculate the shortest vector from the point to each corresponding line. Add these vectors and add the square of the length of the resultant to the cumulative sum. The threshold is set to 0.8 so that only points for which the corresponding lines are fairly certain contribute to the measure. The cumulative sum is normalised by the number of pixels which have contributed.

Note that this measure will not, in general, give zero for identical sets of images. The reason for this is that the corresponding line will not necessarily pass through the point which is being matched (see chapter 7). An additional problem with this measure is that it is not independent of the mechanism used to generate the reconstruction. However, in experiments described below, we will show that it is more suitable than correlation for comparing the qualities of two reconstructions. This new measure will be referred to as the correspondence alignment measure (CAM).

Finally we note that both the CAM and correlation give values for individual sections. In order to compare reconstructions as a whole we will compare the mean and standard deviation of these individual values over the whole stack.

9.2 Analysis of the RD

The RD has been proposed above as a measure for comparing the differences between two warps. In this section we give formulae to indicate how the mean and standard deviation of the RD behave as an image is transformed by rotation, uniform scaling, and scaling in the x-axis. Formulae for translation are not given because clearly the displacement for each point is identical and equal to the modulus of the translation vector.

To calculate the formulae, we assume that the origin coincides with the centre of the image and for simplicity, that the side lengths of the image are divisible by two. Let the half side lengths of the image be a and b, then the mean, m, and standard deviation, σ, of the RD for each transformation are:

rotation:

$$m = \frac{2\sin\frac{\theta}{2}}{ab}\sum_{x=0}^{a}\sum_{y=0}^{b}(x^2+y^2)^{\frac{1}{2}},$$

$$\sigma = \frac{2\sin\frac{\theta}{2}}{ab}\left[ab\sum_{x=0}^{a}\sum_{y=0}^{b}(x^2+y^2) - \left(\sum_{x=0}^{a}\sum_{y=0}^{b}(x^2+y^2)^{\frac{1}{2}}\right)^2\right]^{\frac{1}{2}},$$

where θ is the angle of rotation.

uniform scaling:

$$m = \frac{|s-1|}{ab}\sum_{x=0}^{a}\sum_{y=0}^{b}(x^2+y^2)^{\frac{1}{2}},$$

$$\sigma = \frac{|s-1|}{ab}\left[ab\sum_{x=0}^{a}\sum_{y=0}^{b}(x^2+y^2) - \left(\sum_{x=0}^{a}\sum_{y=0}^{b}(x^2+y^2)^{\frac{1}{2}}\right)^2\right]^{\frac{1}{2}},$$

where s is the scale factor.

scaling in the x-axis:

$$m = \frac{|s_x-1|}{a}\sum_{x=0}^{a}x,$$

$$\sigma = \frac{|s_x-1|}{a}\left[a\sum_{x=0}^{a}x^2 - \left(\sum_{x=0}^{a}x\right)^2\right]^{\frac{1}{2}},$$

where s_x is the scale factor for the x axis, and a is the half length of the image in the x direction.

From these formulae we see that under rotation and scaling the mean and standard deviation of the RD depends on the size of the image, and that, when the same image is considered, they vary linearly under scaling and according to the sine of half the angle under rotation. Clearly, the mean of the RD also varies linearly under translation. Therefore we conclude that the RD is suitable for comparing two warps as long as these warps were calculated for images with identical domains.

9.3 Experiments to Test and Analyse the CAM

The performances of correlation and the CAM cannot be evaluated analytically and two sets of experiments were designed to analyse their performances. The first set was designed to test and evaluate the performances of correlation and the CAM; the second set was designed to see how the CAM changes as a section is transformed. In all cases the images used were those from the 9 day mouse embryo used in chapter 4.

In some of the experiments described below, sections are warped using the warping algorithm described in chapters 5 to 8. The details of the implementation of this algorithm are given in chapter 10, and in all cases iteration scheme 4 of chapter 8 was used, which was to warp alternate sections using the most recent information.

9.3.1 Comparison of Correlation and CAM

Experiment 1

Correlation and the CAM values were calculated for a registered stack of images, and a warped stack of the same images which is clearly better in terms of visual criteria. The stack used was a stack of 50 alternate sections chosen from the middle of the 9 day mouse embryo series and registered using Hibbard's method. A section through the registered and warped stacks is shown in figure 9.1. Note that the warping algorithm has had the required effect on the 9 day series in that the boundaries of structures are more clearly defined.

Results

Figure 9.1: The results of warping 50 sections of the 9 day series. The top image shows the registered stack; the bottom image shows the results of warping the registered stack.

	1 - correlation		CAM	
	mean	std	mean	std
registered	0.205	0.041	11.120	4.780
warped	0.171	0.026	5.190	0.673

Table 9.1: Table showing the results of experiment 1.

The results of calculating correlation and the CAM for both the registered and warped stacks are given in table 9.1 and figure 9.2 which shows the values of $(1 - correlation)$ and the CAM for each section in the registered and warped stacks. From the graphs and the table we see that the values for the warped stack have lower means and vary less than those for the registered stack although the values for isolated sections may be worse. The difference in the mean and standard deviation is most noticeable for the CAM. In fact the mean values have dropped by 15.3% and 50.6% for $(1 - correlation)$ and the CAM respectively. This shows that both correlation and the CAM may be suitable measures for measuring the quality of the reconstruction.

Experiment 2

This experiment is designed to show that the mean correlation value is not suitable for measuring the quality of a reconstruction. Five images were chosen at intervals of approximately 25 sections from the middle of the 9 day series. A test stack of three sections was generated from each of these sections, and in each stack the first two images

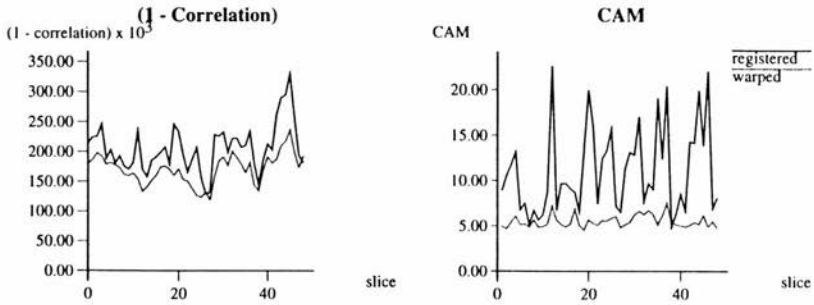


Figure 9.2: Graphs showing the values of $(1 - correlation)$ and the CAM for a registered and warped stack of sections.

were identical and the third section was chosen so that it was adjacent but one in the series to this image. Correlation and the CAM were evaluated for each stack before and after the middle section was warped using the warping algorithm.

Results

The results of calculating correlation and the CAM values before and after the middle section of each test stack is warped are given in table 9.2 which shows the mean and standard deviation of the $(1 - correlation)$ and CAM values for each stack before and after warping. We see that in all cases the mean has increased, but the standard deviation has decreased, and that in all cases the values of the CAM have decreased, as expected, showing that the overall alignment has improved.

	1 - correlation				CAM	
	registered		warped			
	mean	std	mean	std	registered	warped
stack 1	0.151	0.213	0.209	0.094	14.932	8.766
stack 2	0.188	0.266	0.271	0.197	20.260	12.423
stack 3	0.219	0.310	0.299	0.205	16.310	14.263
stack 4	0.179	0.253	0.302	0.113	15.402	8.386
stack 5	0.215	0.303	0.288	0.167	23.075	11.336

Table 9.2: Table showing the correlation and CAM values for the 5 test stacks of experiment 2 before and after warping. The correlation shown is $(1 - correlation)$.

From this experiment we conclude that the mean ($1 - correlation$) value is not a reliable measure as it decreases when a section resembles one of its adjacent sections more than the other. However the standard deviation may still be a suitable measure.

Experiment 3

This experiment was to investigate the standard deviation of the correlation values as a measure of the quality of a reconstruction. A stack of three identical sections was generated and two further stacks were made by

1. translating the middle section by 2 pixels in the x direction, and
2. translating the middle section by 5 pixels in the x direction.

Correlation and the CAM measures were calculated for the latter two stacks.

Results

The results of experiment 3 for the two stacks where the middle section was translated are given in table 9.3 and we see that although the mean ($1 - correlation$) value increases as the sections are translated, the standard deviation remains constant. (It is zero because the correlations between the middle section and the top and bottom sections are equal, because the top and bottom sections are identical.) The CAM values show that the quality of the reconstruction decreases as the middle section is translated.

	1 - correlation		
translation	mean	std	CAM
2 pixels	0.183	0.0	5.939
5 pixels	0.254	0.0	24.376

Table 9.3: Table showing the correlation and the CAM values for stacks of three identical images where the middle section was translated. The value of the CAM for perfect alignment is 1.481.

From these experiments we conclude that the neither the mean nor the standard deviation of correlation is a suitable measure for evaluating the quality of a reconstruction,

whereas the CAM exhibits the required behaviour and is therefore a more suitable measure.

9.3.2 Response of the CAM to Section Transformation

Experiment 4

Five stacks of three identical images were made from the series of five images taken at intervals of approximately 25 sections described in experiment 2. The middle sections were transformed in various ways and the CAM was evaluated. The transformations were:

Experiment 4.1 Rotate the middle section by -10 to 10 degrees at intervals of 1 degree.

Experiment 4.2 Translate the middle section by -10 to 10 pixels at intervals of 1 pixel in both x and y, and in x only and y only.

Experiment 4.3 Scale the middle section by a factor of 0.7 to 1.3 in intervals of 0.025 in both x and y, and in x only and y only.

For the rotation and scale transformations, the centre points of the images remained fixed.

Results

The effects of rotation, translation and scaling on the CAM for all five images are shown in figures 9.3 to 9.5. When the images are rotated (figure 9.3), the CAM varies quadratically up to about five degrees when the slopes of the curves decrease and the differences between the values for the different images become greater. This behaviour for larger angles is due to false correspondences occurring because the correct correspondence is no longer the closest good match. Similar effects are observed for translation and scaling. Lower magnitudes for translation in the x direction, in particular, are due to the fact that a lot of the edges in the images are approximately parallel to the x-axis because the object of interest was captured so that its long axis was roughly (within

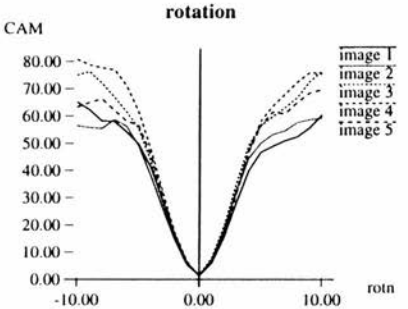
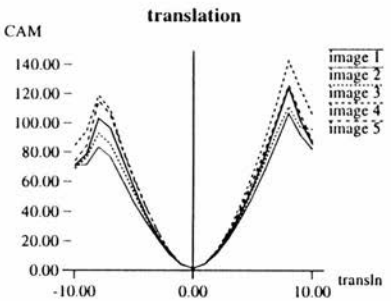
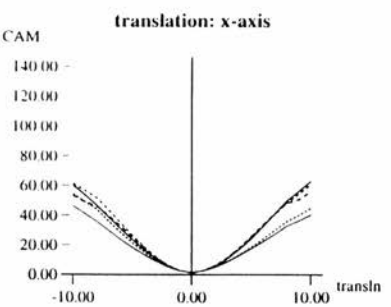


Figure 9.3: Graph showing the effect of rotation (in degrees) on the CAM (in pixels^2).





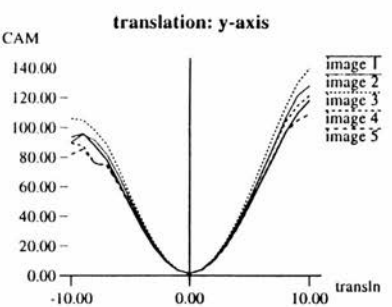


Figure 9.4: Graphs showing the effect of translation (in pixels) on the CAM (in pixels^2).

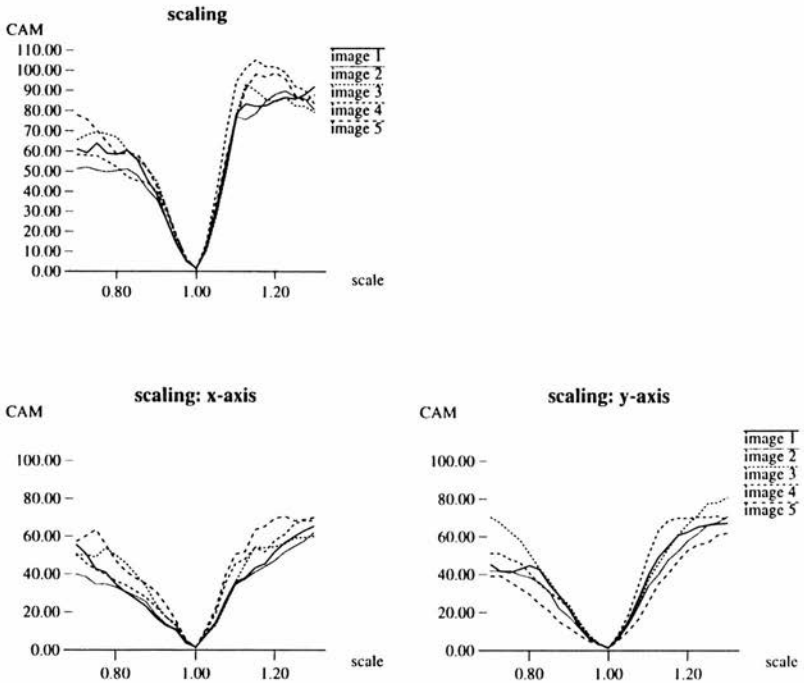


Figure 9.5: Graphs showing the effect of scaling on the CAM (in pixels^2).

15 degrees) parallel to the x-axis, and to the fact that correspondences are allowed to slide along lines with no penalty.

From these results we conclude that the CAM is suitable when the image transformations are small, as the values for all the images are similar, but is unreliable for large transformations because of the high likelihood of false matches. As the differences in position and orientation of images can be assumed to be small, because for warping we may assume that the sections have been registered, we conclude that it is suitable for objectively comparing the quality of a reconstruction before and after warping.

9.4 Summary and Conclusion

In this chapter we have discussed criteria for evaluating the results of experiments for testing the warping algorithm which will be performed in chapter 10. The mean and standard deviation of the RD was proposed and found to be suitable for analysing differences between two warps. In order to measure the quality of a reconstruction, two measures were tested: correlation and a new measure, the correspondence alignment measure or CAM, which is calculated using both adjacent sections. It was found that correlation is unsuitable for this purpose because the mean may increase when a section resembles one adjacent section more than another, and because the standard deviation remains constant when a section is misaligned in the same way with respect to both adjacent sections. When the performance of the CAM was analysed with respect to transformed images, it was found that it is suitable for measuring the quality of reconstructions resulting from registered and warped stacks of images.

Chapter 10

Experiments and Results

In this chapter we test the warping algorithm described in chapters 5 to 8. The aim of the experiments is to determine how the various parameters affect the final result, and to determine how the algorithm can be improved. In addition we examine the displacement maps produced by the warping process to evaluate how accurately they can be approximated by a global warp transformation.

The parameters to test include:

1. the strength of the springs given certain material properties,
2. the effect of changing the search area for calculating correspondences,
3. the number and spacing of the "interesting" points,
4. the resolution of the finite element mesh, and
5. convergence parameters.

A single set of data — the 9 day mouse embryo series used in chapter 4 — will be used to evaluate how the above parameters affect the result, but the performance of the warping algorithm will also be tested on other sets of data, all of which show that the primary objective of this work, to be able to remove sectioning distortions, has been achieved.

10.1 Implementation Details

Before discussing the experiments we first describe some specific implementation details of the algorithm, giving attention to the description of the various parameters. The warping program has several parameters which can be divided into four categories: image, matching, warping and convergence parameters. The parameters, along with their default values, are discussed below. In each case the default values are given in bold type enclosed in square brackets after the name of the parameter. Some of the parameters were determined experimentally while the various algorithms were being developed; the default values of others were chosen arbitrarily, and some of these are the subject of the experiments discussed below.

10.1.1 Image Processing Parameters

The image processing parameters are:

toggle extract-embryo [on] This procedure uses image processing techniques to locate the embryo in the image. It was designed specifically for the 9 day mouse embryo series where all the objects are close to each other, relative to their sizes and the size of the image, and are fully contained within the image. This procedure should be used whenever possible as it reduces the size of the image, and thus the processing time.

The algorithm is as follows:

1. Threshold the image. This is done automatically using a chords analysis of the histogram: the chord joining the histogram peak and the histogram right hand end point is constructed, and the histogram point furthest perpendicularly from the chord is chosen as the threshold.
2. Clean away scraps and thin regions in the image by erosion followed by dilation with a structuring element of radius two (*opening* operator).
3. Join on bits which are close to the embryo by dilation followed by erosion with a structuring element of radius two (*closing* operator).

4. Extract the largest region and dilate it with a structuring element of radius 15 to make sure that the whole of the embryo is included.

5. Calculate the convex hull of the region and fill in pixel values so that the region fills the convex hull. This gives the required image.

Note that all the parameters in this algorithm were selected by observation for the 9 day data set only, but have proved adequate for all image sets tested.

width [1.0] This is the half-width at half-maximum of the derivative of Gaussian used for calculating the gradient image, wliich is part of the information used to choose the nodes of the finite element mesh and the interesting points (see chapter 5).

mask [2] This is the size parameter which determines the width of the square neighbourhood of size $2 \times$ **mask** $+ 1$ used for calculating the features for every pixel in the images (see chapter 7).

10.1.2 Matching Parameters

The matching parameters include variables in the match function given in chapter 7. These remained constant for all the experiments and are not given here. Most of the other parameters are described in chapter 8 and are required for the calculation of the correspondences. The default values for **areasize** and **pointdist** were determined arbitrarily and are tested in experiments described below; the choice of the default value of **power3** were discussed in chapter 8.

areasize [10] The parameter which determines the size ($= 2 \times$ **areasize** $+ 1$) of the search area for the correspondences. The match values for the matchmap are calculated over this area.

pointdist [5] The minimum distance between the interesting points. This is to ensure that the interesting points are scattered evenly over the object of interest (see chapter 5).

tsize [2] This specifies how far the interesting point is moved in each direction (in increments of one pixel) when the scatter of tentative corresponding points is calculated (see chapter 8).

power3 [10] The half-width at half-maximum of the Gaussian used for converting the sum of the squared distances of the tentative corresponding points from the corresponding line to a spring constant (see chapter 8).

10.1.3 Warping Parameters

The warping parameters are those used for calculating the finite element mesh, and those which describe the mechanical properties of the mesh. These are:

mindist1 [7] The minimum spacing of the nodes of the finite element mesh for the part of the mesh covering the object.

mindist2 [11] The minimum spacing of the nodes of the finite element mesh for the part of the mesh covering the background.

springk [10] The constant used to multiply the calculated spring constant to make it compatible with the material properties.

type [3] The type of the finite element mesh. The options implemented are:

1. Linear
2. Non-linear incompressible
3. Non-linear compressible

Experiments were only performed on type 3 elements. Linear properties were not used since, as discussed in chapter 6, they are unsuitable for our purposes. Although incompressible material properties would allow the material to be compressed (by causing the section to thicken), a material with Poisson's ratio equal to zero was preferred so that the effects of each force applied to the sections were less affected by the effects of the other forces.

material properties These were set arbitrarily to give the strain energy function

$$W = \lambda_1^{4.5} + \lambda_2^{4.5} - 2 - (\lambda_1^{-2.0} + \lambda_2^{-2.0} - 2)$$

(see chapter 6). This function describes a material for which the initial (that is for zero strain) elastic modulus is 6.5.

The default value for **springk** was set arbitrarily; those for **mindist1** and **mindist2** were set by visual examination of the finite element mesh superimposed on images from the 7 day series. However these three quantities are the subject of experiments described below.

10.1.4 Convergence Parameters

There are three levels of iteration in the warping algorithm and they each require parameters in order to control convergence. Some parameters also ensure that the process does converge.

FEstop [0.2] (finite element iteration) Stop the finite element iteration when the maximum displacement calculated during a single iteration is less than this value. Note that there is no check that the finite element calculations are converging since it was assumed that as long as the material properties and the spring constants are suited to each other, this would not be a problem.

warpstop [0.5] (force iteration) Stop the force iteration when the maximum displacement calculated during a single force iteration is less than this value.

recalc [5.1] (force iteration) Recalculate the image and the features when the maximum displacement since the last time the image was recalculated exceeds this value.

maxiters [11] (force iteration) Maximum number of iterations for the force iteration.

stopcondition [0.8] (stack iteration) Warping is deemed to have converged when the maximum displacement for a stack iteration is less than this value.

maxiteration [40] (stack iteration) Maximum number of iterations for the stack iteration.

The default values for these parameters were all determined arbitrarily, and all, except **maxiteration**, which was set to ensure that the warping process does stop, are tested in an experiment described below. However, some conditions can be imposed. For

instance, for convergence it is clearly necessary that **FEstop** is less than **warpstop** and that **warpstop** is less than **stopcondition**. In addition, as discussed in chapter 6, **FEstop** should be sufficiently small to make sure that the forces in the FE system are in equilibrium when the springs are recalculated. If this is not the case the calculation may become unstable.

10.1.5 Other Implementation Details

To reduce the space required for processing, the warped images are not stored but are recalculated along with the feature measurement vectors each time before they are used. The warped images are always calculated from the originals to prevent successive deterioration of the images. To speed up the calculations, sections were not warped if the maximum displacement for their adjacent sections was less than 0.5 pixels since the last time they were warped. This is to prevent oscillation caused by small changes to the images each time they are recalculated. Although the displacements are represented as floating point numbers, a similar precision is not obtained when the images are calculated because nearest neighbour values are used. This can cause displacements of less than 0.5 pixels to lead to changes in the images because it is not possible to move any point in an image by less than one pixel. When this rule was lifted, oscillation was observed for experiments on stacks of three sections when the warping process failed to converge after two iterations. When the feature attributes of the interesting points were kept, and not recalculated, the warping process converged as expected. The feature attributes could not be kept in the case of stacks of several sections because of the amount of space required to store them.

The results of the warp are written to disc in a file which contains (for each section)

1. the number of elements in the finite element mesh,
2. the number of nodes in the finite element mesh,
3. a list of the elements in the finite element mesh,
4. the nodal displacements,
5. a list of the interesting points,

6. the current section,

7. the current iteration, and

8. the maximum displacement so far in the current iteration.

The last four items are included to give enough information to restart the program should it be stopped. The program was written so that if it received an interrupt signal it would write the above information to disc before stopping. As the number of interesting points has a direct effect on the processing time, the number of interesting points was limited to 500. This should have no effect on warping, because when interesting points are found, points on edges are picked first followed by points in regions of texture. By limiting the number of interesting points, only points in areas of texture are discarded. Note that the slowest parts of the calculation are the calculation of the feature attribute vectors and matching. An analysis of the costs of the warping process and a description of other aspects of the implementation are given in appendix D.

10.2 Methods

In this section the experiments for testing the various parameters are described. As discussed in chapter 9, two types of experiment are performed: The effects of some parameters such as those controlling the behaviour of the matching and warping parts of the algorithm can be evaluated using test images obtained by manually warping images from the series. The effects of the other parameters, and of warping as a whole, have to be evaluated from the results of warping a stack of sections.

The aim of the experiments is to test the warping algorithm with respect to performance and robustness. By testing the algorithm with respect to performance we mean checking that the results obtained by the warping process meet our objectives with respect to deformation correction and smoothness in the reconstruction. That is, that the resulting reconstruction should be smooth and be a reasonable representation of the originally sectioned object in that all the parts have the correct relationship to each other, though the reconstruction need not be identical to the originally sectioned object. By testing the algorithm with respect to robustness, we mean checking that it produces satisfactory

results for a wide range of sections, and that small changes in the parameters do not lead to large changes in the results.

To evaluate the results of the experiments on test images, we will analyse the RD as described in chapter 9. The results of experiments on stacks of images will be evaluated using visual criteria and the CAM, also described in chapter 9.

We now describe the experimental methods and the experiments. All experiments on the 9 day mouse embryo series were performed on images where the embryo had been extracted (section 10.1.1).

10.2.1 Experimental Methods

Test Images

We first describe the experimental method for the use of test images in detail. Most experiments involving test images were performed on stacks of three identical sections where the top and bottom sections were unmodified and the middle section was deliberately warped[1]. These stacks will be referred to as *AWA stacks*.

The warped images in the AWA stacks were made by applying one of four warps to the original image. These warps are illustrated for one image in figure 10.1 and were obtained by manually attaching springs to an image in such a way as to bend the image along its horizontal axis, both up and down (warps 1 and 2), and to stretch the image in either the x or y direction (warps 3 and 4). Statistics of the pixel displacements of the warps are given in table 10.1. We note that warps 3 and 4 cause larger displacements than warps 1 and 2.

20 AWA stacks were made by choosing five images at intervals of approximately 25 sections from the middle of the 9 day series and applying the four warps to each image. The warps are valid for all five images as the domain of the warps covers the area containing the object in all these images.

In experiment 4, these five images chosen from the series were also used to make other

[1] The warping algorithm was also tested on a stack of identical, unmodified images. In this case the maximum displacement calculated was insignificant (0.2 pixels)

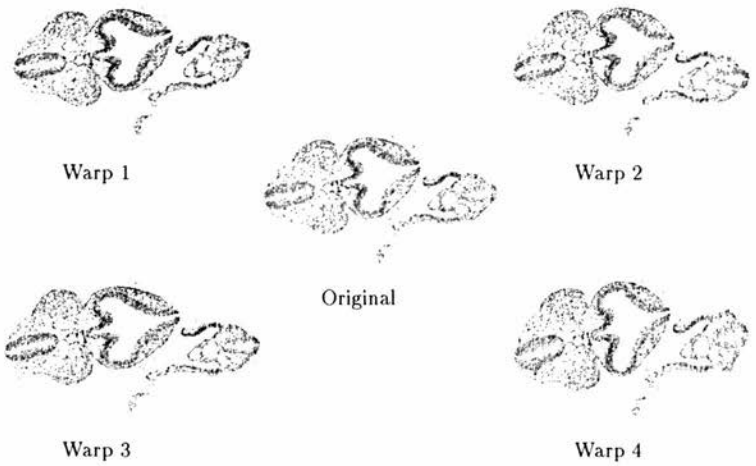


Figure 10.1: The four warps applied to one of the test images.

	x displacement			y displacement			Euclidean distance		
	max	mean	std	max	mean	std	max	mean	std
warp 1	7.14	1.89	1.89	24.48	7.25	5.40	24.48	7.67	5.45
warp 2	8.31	2.12	1.90	12.65	8.05	3.27	12.66	8.77	4.75
warp 3	30.20	14.54	8.21	4.39	1.18	0.91	30.20	14.66	9.31
warp 4	6.31	1.89	1.22	34.53	15.99	6.56	34.54	16.19	9.29

Table 10.1: Table showing the displacement statistics for the four artificial warps used to make test images.

stacks of 3 images: *ABA stacks*. In this case the middle section was chosen to be one of the sections adjacent but one in the series to the (identical) sections comprising the top and bottom sections.

A Stack of Sections

For the experiments requiring a stack of sections (experiments 1, 5 and 6), 50 alternate sections were selected from the middle of the 9 day mouse embryo series and were registered using the algorithm described by Hibbard *et al.*. Alternate sections were

chosen to increase the range of variation in the image content. We will refer to this stack as the *50 section stack*.

10.2.2 Experiments

Experiment 1: Stack Iteration

In the first experiment the two most promising options from the iteration schemes discussed in chapter 8 were tested. The aim of this experiment was to choose one iteration scheme which would be used for all the other experiments. The two iteration schemes tested were:

scheme 1 Split the stack interation into two passes: the first warping all the odd numbered sections; the second all the even numbered sections. For each pass use information from the previous pass through the stack.

scheme 2 Warp each section consecutively using the most recent information and al ternate the direction of warping through the stack.

To test the two schemes, a stack of 10 identical sections was generated, various sec tions were transformed, and the stack was warped using both iteration schemes. The transformations were:

- Randomly rotate or translate each of the middle 8 sections by up to 5 pixels (in either, or both, directions) or 5 degrees. Four sets of 8 random transforma tions were generated: two involving random rotations, and two involving random translations. We will call the resulting stacks *RT stacks*.
- Rotate or translate the fifth section by 5 degrees or 5 pixels. We will call these stacks *TMS stacks*.

For all of these experiments the "correct" result is known and therefore the calculated warps can be measured against the original transformations using the RD.

In addition, the 50 section stack was warped using both iteration schemes to see how their performance compares on real serial section images. Note that the evaluation of

this part will consist of comparing the warp transformations and evaluating the quality of the reconstruction as a whole using CAM and sections cut perpendicularly to the cutting direction.

Experiment 2: Spring Constant

The spring constant (**springk**) and search area (**areasize**) parameters are related since a larger search distance will give rise to more distant corresponding points which will result in stronger forces being exerted by the springs. It is important to employ appropriate values for the spring constants with respect to the material properties as values which are too low will cause the warp to be limited by the elastic properties, but values which are too high cause convergence problems with the finite element iteration. Although these convergence problems can be eliminated by incrementally increasing the forces, this was not implemented because the material properties should limit the amount of warping. If the forces are too strong with respect to the material properties, an incorrect reconstruction will result as true shape changes are smoothed into a linear progression between the top and bottom sections.

The aim of this experiment was to determine maximum values for the spring constant for each search area, and to test whether high values of **springk** and **areasize** are more successful at warping the image back to the original. For each **areasize** of 5, 10, 15 and 20 pixels, **springk** was set to 0.5, 1, 2, 4, 6, to 20. This experiment was performed on AWA stacks.

Experiment 3: Density of Interesting Points

It is to be expected that the warp will be more accurate if there are more interesting points. To test this **pointdist** was set to 2, 4, 6, to 16 and the AWA stacks were warped. Note that for this experiment the maximum number of interesting points was raised to 5000, effectively removing the restriction, as 5000 points were never found.

mindist1	2	4	6	8	10	12	14	16	18	20
mindist2	4	8	10	12	14	14	14	16	18	20
mindist1	22	24	26	28	30	32	34	36	38	40
mindist2	22	24	26	28	30	32	34	36	38	40

Table 10.2: Table showing the values of **mindist1** and **mindist2** used in experiment 4.

Experiment 4: Mesh Resolution

The finite element method gives numerically more accurate results (with respect to the stress and the strain) when the finite element mesh is finer. To see what effect this has on the calculated warp, **mindist1** and **mindist2** were tested on AWA stacks with the values shown in table 10.2. From the table we see that **mindist2** is greater than **mindist1** for values up to **mindist1** = 14. This was allow convergence of the finite element method for these higher resolutions.

Since the warped images in the AWA stacks have undergone a global warp, this experiment was repeated on the ABA stacks.

Experiment 5: The Convergence Process

To show how the warping process converges and how the stack changes at each iteration, a stack of the first 40 sections of the 50 section stack (the *40 section stack*) was warped and at the end of each iteration the warp transformation was written to disc.

Experiment 6: Convergence Parameters

The aim of this experiment is to gauge how much effect the convergence parameters have on the quality of the reconstruction. The 50 section stack was warped with the different values for the convergence parameters given in table 10.3. From this table we note that the first four experiments test the effect of increasing all the convergence parameters, and whether restricting the number of force iterations, which is computationally the most expensive part of the algorithm, has an adverse effect. The other experiments test the effects of varying each of **FEstop** (experiments 6.2, 6.5 and 6.7), **warpstop**

Exp. No.	FEstop	warpstop	maxiters	stopcondition
6.1	0.2	0.5	11	0.8
6.2	0.2	0.5	2	0.8
6.3	0.4	1.0	11	1.6
6.4	0.4	1.0	2	1.6
6.5	0.4	0.5	2	0.8
6.6	0.4	0.75	2	1.0
6.7	0.6	0.75	2	1.0
6.8	0.6	1.0	2	1.2
6.9	0.6	1.25	2	1.6
6.10	0.6	1.0	2	1.6

Table 10.3: Table showing the values of the convergence paramenters for experiment 6.

(experiments 6.5 and 6.7 to 6.9), and **stopcondition** (experiments 6.5, 6.6, 6.8 and 6.10). Note that in some cases more than one parameter increases. This is to ensure that

$$\textbf{FEstop} < \textbf{warpstop} < \textbf{stopcondition}.$$

Experiment 7: Other Image Sequences

The warping algorithm was applied to other sets of data. These series were:

1. a 7 day mouse embryo,
2. a normal mouse embryo heart, and
3. an abnormal mouse embryo heart.

Both hearts were from embryos which were approximately 10 days old. All these series were digitised using a Zeiss Axioplan mircoscope with a x5 objective, a Xillix 1400 12 bit CCD camera, and the MRC Human Genetics Unit S-bus DMA image capture interface running on a SPARC 10/41.

The 7 day embryo was embedded in plastic (araldite), sectioned at 2 microns, digitised at a resolution of 1.3 microns, and resampled by a factor of 3. The warping algorithm was tested on 100 alternate sections taken from the middle of the series. The default

values of all the parameters, except for **mindist1**, **mindist2** and **pointdist** which were 3, 3, and 6 respectively, were used.

Both sets of heart data were perfused (that is, injected with fixative to make them expand), embedded in paraffin wax, sectioned at 7 microns, stained with haematoxylin and eosin and digitised at a resolution of 1.3 microns. The normal and abnormal heart series consisted of 50 and 60 sections respectively. The heart series present an additional challenge because each image represents only part of the 2D section. However the object of interest, the heart, is located near the centre of the images and is fully contained in the images. To avoid edge effects, interesting points were restricted to the largest circle contained within each image to ensure that the corners of the images were not matched. The heart sections were registered using the algorithm described by Hibbard *et al.* with the addition of a Welch window to remove edge effects. The convergence parameters of experiment 6.2 were used and the parameters **mindist1**, **mindist2** and **pointdist** had values 15, 20 and 10 respectively.

Experiment 8: Approximation by Global Transformations

In order to see to what extent the calculated warps can be approximated by global warp transformations, the warp transformations obtained in experiments 6.1 and 8 were approximated by each of three global warp transformations: a rigid transformation (rotation and translation only), an affine transformation, and a polynomial warp of degree two. These transformations were fitted by a least squares approximation based on the nodal displacements of the finite element mesh. The functions fitted to the data are as follows, where $[x, y]^T$ describes the original position of a point and $[x', y']^T$ describes the position of the point in the transformed configuration:

rigid transformation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 x & + & a_2 y & + & a_3 \\ -a_2 x & + & a_1 y & + & a_4 \end{bmatrix}$$

affine transformation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 x & + & a_2 y & + & a_3 \\ a_4 x & + & a_5 y & + & a_6 \end{bmatrix}$$

2nd order polynomial transformation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1x^2 + a_2y^2 + a_3xy + a_4x + a_5y + a_6 \\ a_7x^2 + a_8y^2 + a_9xy + a_{10}x + a_{11}y + a_{12} \end{bmatrix}$$

Note that to obtain a transformation without scale, the parameters a_1 and a_2 in the rigid transformation were rescaled to satisfy $a_1^2 + a_2^2 = 1$. This approach was implemented because in this application the scale is expected to be close to 1. The displacement maps for the rigid, affine and polynomial approximations were each subtracted from the displacement maps generated from the original warp transformations and the results were analysed using the mean and standard deviation of the RD.

10.3 Results

Experiment 1: Stack Iteration

The number of iterations needed for each of the four RT stacks, and the values of CAM for both iteration schemes are shown in table 10.4. Transformation sets 1 and 2 consisted of random rotations; transformation sets 3 and 4 consisted of random translations. Here we see that sometimes scheme 1 converges faster than scheme 2 and sometimes scheme 2 converges faster than scheme 1. Figure 10.2 shows the mean RD for the random transformations for the two schemes along with the difference in the calculated warps, which is shown so that the agreement between the two methods can be compared. The mean displacements of the applied transformations are also shown to give an indication of the original conditions. From these graphs we see that the two iteration schemes do not converge to identical results and that scheme 2 is more successful at correcting random transformations. Examination of the values of the CAM however, shows that, except for random set 1, the quality of the reconstruction is similar for both iteration schemes.

Both schemes were equally good at correcting the single section which had been translated or rotated out of register in the TMS stacks. In both cases the mean residual displacement was less than one pixel, and the same number of iterations were required.

When the iteration schemes were tested on the 50 section stack it was found that, for most sections, the mean difference in displacement was less than 1 pixel (figure 10.3).

	scheme 1			scheme 2		
	I	mean	std	I	mean	std
random set 1	10	3.037	1.064	4	2.513	0.172
random set 2	3	2.365	0.127	5	2.260	0.143
random set 3	4	1.667	0.325	3	2.052	0.362
random set 4	5	1.961	0.277	4	2.079	0.383
50 sections	6	5.190	0.673	14	5.571	0.750

Table 10.4: Number of iterations (I) and values of CAM (pixels^2) for both iteration schemes applied to the 4 RT stacks and the 50 section stack.

We also see that the visual quality of the two reconstructions is similar (figure 10.4), but from the last line in table 10.4 we see that iteration scheme 1 produced a better reconstruction than scheme 2, as measured by the CAM, and converged must faster.

From the results for randomly transformed sections, we conclude that whilst neither iteration scheme perfectly corrects the rigid transformations applied to the images, both (as can be seen from the values of the CAM) achieve a smooth reconstruction. In principle, both iteration schemes should be able to correct these global transformations because the outer sections are fixed. However, convergence may be slow because a large global transformation in one section will propagate cooperatively to neighbouring sections and then only slowly be reduced. The slower convergence of scheme 1 with respect to random set 1 almost certainly arises from this effect and the chance that the first pass may warp both adjacent sections to a section with a large transformation, which will then require more iterations to correct. Note that, in this case, the convergence process will be slow because the correcting displacements will only be halved at each iteration.

From the above we conclude that the warping algorithm cannot be relied upon to satisfactorily correct large registration errors, but it will produce smooth surfaces whatever the starting point. Therefore the best results will be obtained by registering the sections prior to warping (which is also sensible due to the computational cost of warping).

We conclude that both schemes give satisfactory reconstructions, and we choose scheme 1 because, it yields a significantly better value of the CAM for the stack of 50 sections, and converges much more quickly in this case.

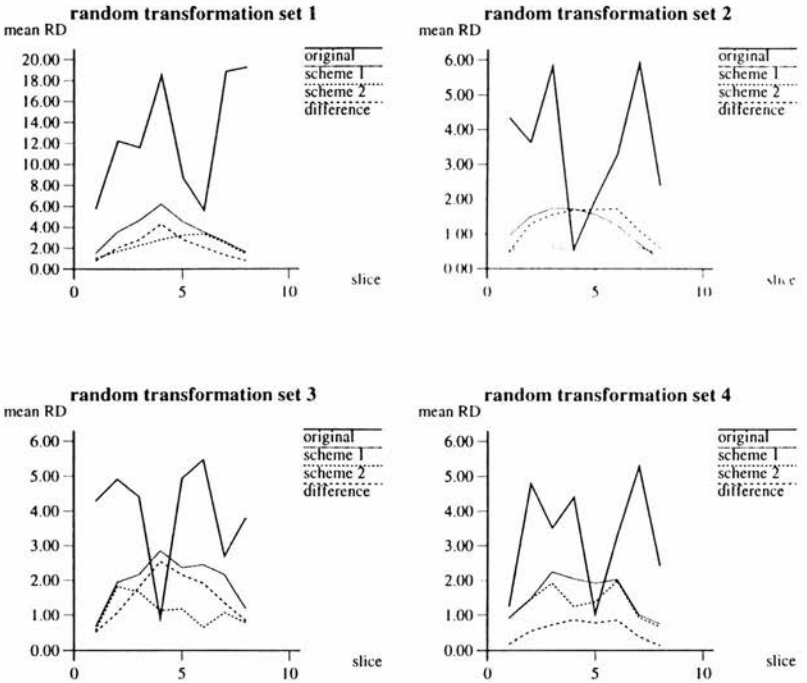


Figure 10.2: Graphs showing the mean RD (in pixels) for scheme 1 and scheme 2 applied to the RT stacks. The mean displacements of the difference between the two calculated warps are also displayed.

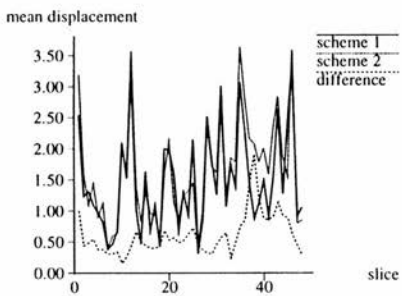


Figure 10.3: Graph showing the mean displacement (in pixels) of the warps calculated by the two iteration schemes for the 50 section stack. The difference in the warps is also shown.

registered

scheme 1

scheme 2

Figure 10.4: The results of applying scheme 1 and scheme 2 to the 50 section stack.

10.3.1 Experiment 2: Spring Constant

The results of varying **springk** and **areasize** are given in figures 10.5 and 10.6. Figure 10.5 shows the results of warping all the AWA stacks, keeping **areasize** constant at 10 pixels. The results for the other values of **areasize** are not shown, but the graphs gave similar information. The results of varying **areasize** are shown in figure 10.6 which shows the results of warping all the AWA stacks made from image 3. Again, the results for the other images were similar.

From figure 10.5 we see that the mean RD decreases as **springk** increases, and that the resulting curve is, in most cases, hyperbolic giving a sharp decrease for lower values of **springk** but flattening out for higher values.

From figure 10.6 we see again that, as **springk** increases, the mean RD decreases and that this is true for all values of **areasize**, except for **areasize** = 5 for warp 4. For warps 1 and 2 we see that, for all values of **springk**,there is little difference in accuracy in the results of varying **areasize**. However for warp 3, the results for **areasize** = 5 are

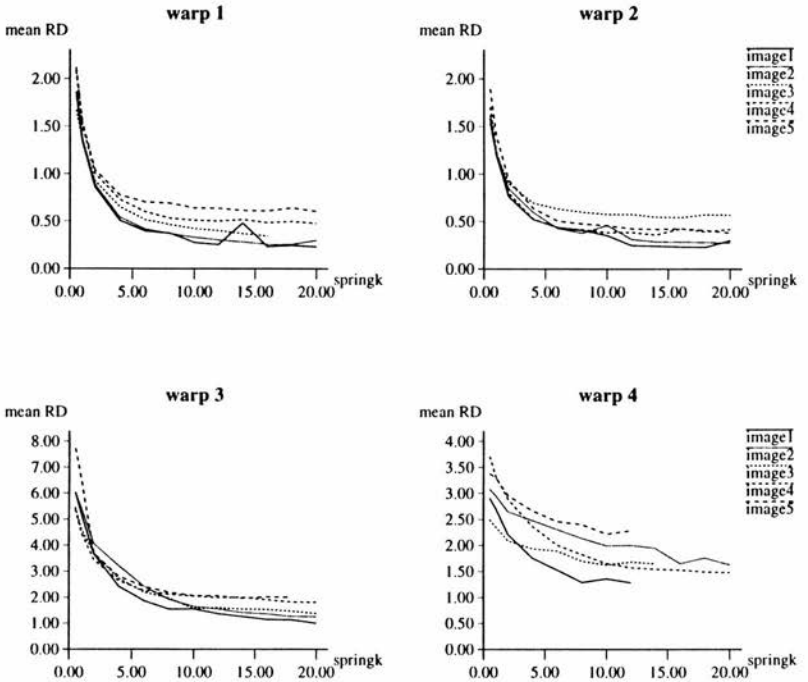


Figure 10.5: Graphs showing the results of warping all 20 AWA stacks keeping **areasize** constant at 10 pixels and varying **springk**. Note that for some images, the process failed to converge for the highest values of **springk**. The mean RD is shown in pixels; the units for **springk** are N/pixels.

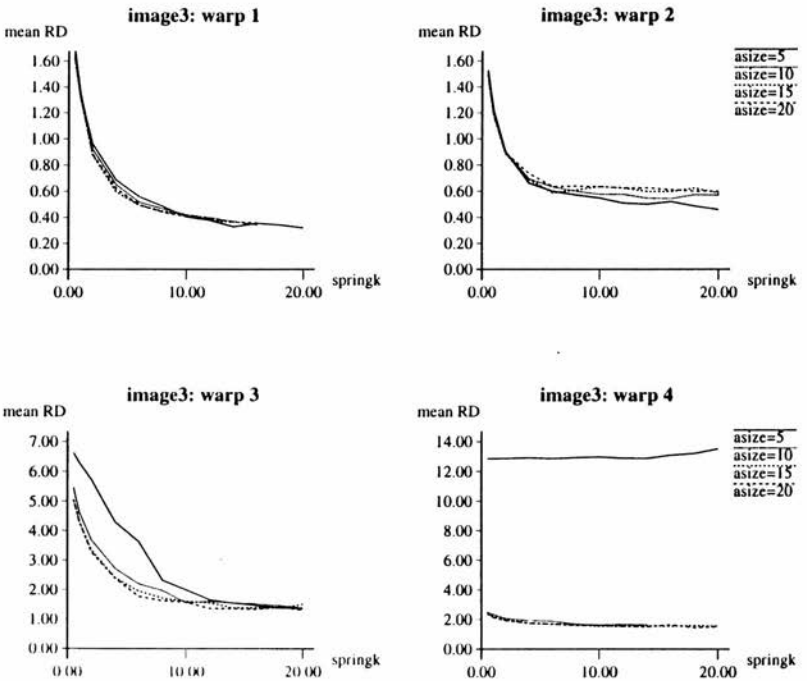


Figure 10.6: Graphs showing the results of the effects of varying **areasize**. The mean RD is shown in pixels; the units for **springk** are N/pixels.

poor for low values of **springk**, but for warp 4 the results for **areasize** = 5 are much worse than the results for the other values of **areasize** for all values of **springk**. This suggests that for all warps a minimum **areasize** exists for which satisfactory results will be calculated, and that increasing **areasize** over this value has no effect on the accuracy of the results. We note also that **areasize** can be less than the maximum displacement required since as we can see from table 10.1, the maximum displacement for the warps was up to 35 pixels, but choosing **areasize** = 10 is sufficient to bring the images into alignment.

We conclude from these results that increasing **areasize** over a certain threshold has no effect on the results. Although increasing **springk** does have a positive effect on the results, above a certain value (about 10 for these images) large increases in **springk** are required to significantly improve the alignment of the images. However, we do not require such large forces as these would cause biological shape changes to be smoothed away and therefore, we conclude that the convergence properties of the finite element method, as implemented here, are adequate for our purposes.

10.3.2 Experiment 3: Density of Interesting Points

The results of varying the minimum distance between interesting points while warping AWA stacks, are given in figure 10.7. From these graphs we see that the mean RD increases as the minimum distance between the interesting points increases. Furthermore, for warps 1 and 2, and for the lower values of **pointdist** for warps 3 and 4, this relationship is linear, but as the values of **pointdist** increase, significantly larger errors are observed for some images and some values. In addition, we note that for the very lowest values of **pointdist** (up to 6), the finite element method occasionally failed to converge. This is indicated by the missing points in the graphs. Since it is computationally expensive to match large numbers of interesting points, we conclude that **pointdist** should be chosen to ensure sufficient accuracy and for these images a value of **pointdist** = 7 is appropriate.

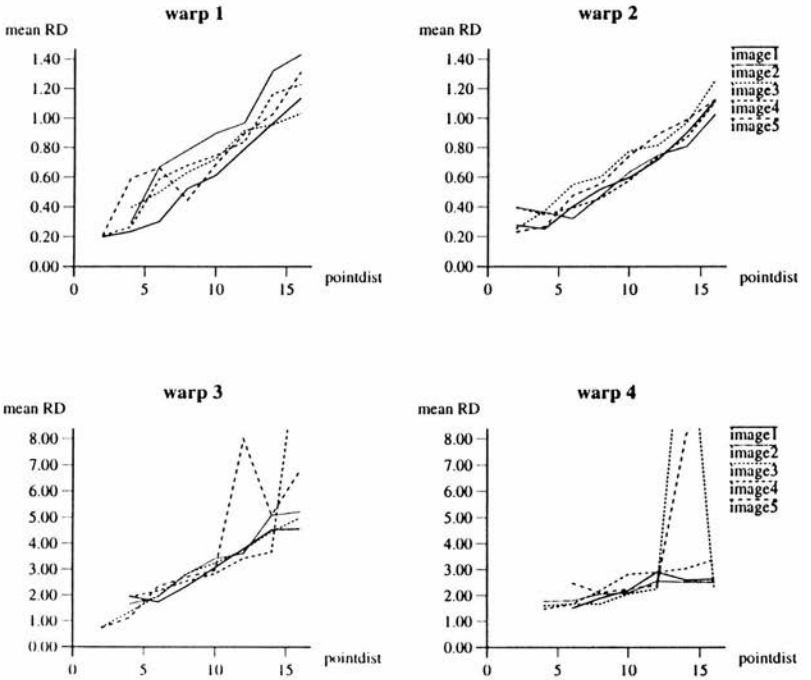


Figure 10.7: Graphs showing the results of the effects of varying **pointdist**. Both the mean RD and **pointdist** are shown in pixels.

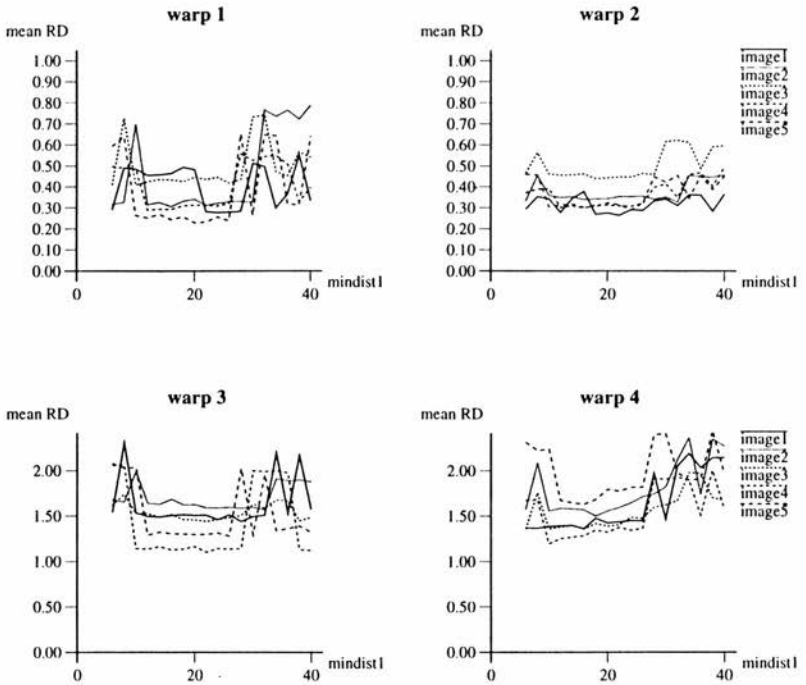


Figure 10.8: Graphs showing the results, for the AWA stacks, of the effects of varying the resolution of the finite element mesh. Both the mean RD and **mindist1** are shown in pixels.

10.3.3 Experiment 4: Mesh Resolution

Figures 10.8 and 10.9 show the results of varying the resolution of the finite element mesh for both the AWA and ABA stacks. From these graphs we see that the resolution of the mesh has no strong systematic effect on the accuracy of the result, although the CAM does increase significantly with respect to the mesh resolution over the test range. This is surprising as we would expect the accuracy to decrease as the resolution of the mesh decreases because coarser resolutions limit local warping. However, the results shown in figure 10.9 show that the mesh resolution makes little difference when warping (nearly) adjacent sections.

These results are affected by the fact that since the material properties are uniform

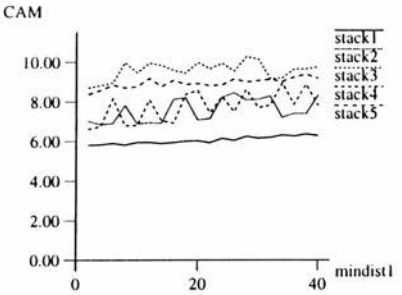


Figure 10.9: Graphs showing the results, for the ABA stacks, of the effects of varying the resolution of the finite element mesh. The CAM is shown in pixels^2; **mindist1** is shown in pixels.

iteration	mean	std
registered	11.119523	4.779772
iteration 1	5.720964	0.784895
iteration 2	5.595593	0.705386
iteration 3	5.537617	0.679738
iteration 4	5.547075	0.682444

Table 10.5: Table showing the values of the CAM (pixels^2) after each iteration performed for experiment 5.

throughout the section, it does not matter whether the boundaries of the elements follow the image contours or not, because the positions of the attachment of the springs are not limited to the node positions and can be attached anywhere within an element. This means that all springs will influence the result whatever the resolution of the mesh. However, this argument only accounts for the results obtained for the global warps. It does not account for the possible loss of local warping due to larger elements in the ABA stacks and we therefore conclude that the required warps, in this case, must vary over longer distances than the mesh size.

10.3.4 Experiment 5: The Convergence Process

The results of calculating the CAM as the 40 section stack is warped are given in table 10.5 and figure 10.10 shows a section in the same x-z plane of the stack after each iteration. By examining this figure we see that there is little discernible improvement

in the reconstruction (in this particular section) after the first iteration. Table 10.5 shows that except for the last iteration, the alignment of the images improves with each iteration. Although the results for the last iteration seem to indicate a slight deterioration in quality of the reconstruction, this deterioration is not significant and is probably due to the interpolation algorithm used (nearest neighbour) for calculating the warped images.

0 iterations

1 iteration

2 iterations

3 iterations

4 iterations

Figure 10.10: The results after each iteration of warping the 40 section stack.

The statistics for the displacements calculated at each iteration, along with the total

iteration	max	mean	std
iteration 1	10.397	1.135	0.842
iteration 2	4.488	0.322	0.253
iteration 3	1.315	0.126	0.110
iteration 4	0.452	0.009	0.026
total displacements	10.206	1.328	0.929

Table 10.6: Table showing the statistics in the displacements (in pixels) calculated at each iteration for experiment 5. Note that the mean and standard deviation values for iteration 4 are misleading because few sections were warped during this iteration.

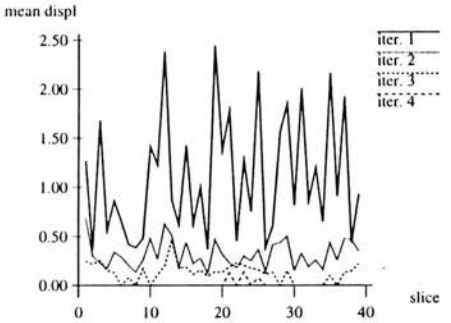


Figure 10.11: Graph showing the mean displacement (in pixels) for each iteration for experiment 5.

displacement after the last iteration, are shown in table 10.6. Here we see that the maximum and mean displacements and standard deviation of the displacements all decrease at each iteration. We note also that the mean and standard deviation of the total displacements are slightly higher than those after the first iteration. Figure 10.11 shows the mean displacement for each section for each iteration. We see that the mean displacements decrease for each iteration, and that the most dramatic decrease occurs after the first iteration. Although the displacements are too small to easily discern differences in the images in figure 10.10, the CAM indicates an improvement in the results.

The above results show that the reconstruction improves with each iteration. Since the mean displacement is so small for the second and subsequent iterations that no visually discernible difference in in the reconstruction is observed, we may also draw the

	parameters					CAM	
Exp. No	**FEstop**	**Wstop**	**MaxI**	**Cond**	I	mean	std
6.1	0.2	0.5	11	0.8	6	5.51285	0.72371
6.2	0.2	0.5	2	0.8	5	5.56171	0.84468
6.3	0.4	1.0	11	1.6	4	5.72878	0.94099
6.4	0.4	1.0	2	1.6	3	5.78838	0.94190
6.5	0.4	0.5	2	0.8	5	5.55191	0.79494
6.6	0.4	0.75	2	1.0	5	5.75163	0.95457
6.7	0.6	0.75	2	1.0	4	5.81623	0.96805
6.8	0.6	1.0	2	1.2	4	5.85754	0.98551
6.9	0.6	1.25	2	1.6	3	6.02434	1.11716
6.10	0.6	1.0	2	1.6	3	6.02248	1.10321

Table 10.7: Table showing the number of iterations (I) and resulting mean and standard deviation values of the CAM (pixels^2) for experiment 6. The abbreviated parameters are **Wstop: warpstop, MaxI: maxiters**, and **Cond: stopcondition**.

conclusion that the maximum displacement is not the ideal criterion of convergence. A better criterion would be based on a measure of alignment. However, the correspondence alignment measure, as implemented, would be too slow to use for this purpose.

10.3.5 Experiment 6: Convergence Parameters

The results of warping the 50 section stack with different convergence parameters are shown in table 10.7 which shows the number of stack iterations to convergence and the CAM for each experiment. We examine these experiments in groups. First we consider experiments 6.1 to 6.4. From these we see that relaxing the convergence parameters has more effect on the result than restricting the number of iterations in the force calculation. We note also that restricting the number of force iterations causes an increased rate of convergence since there are fewer stack iterations and fewer force iterations for each section. It is interesting that restricting the number of force iterations only produces a small deterioration in the result and increases the rate of convergence. This shows that it is not essential that the force iteration is continued to convergence, but on the contrary that performing only a few iterations increases the rate of convergence significantly without significantly changing the results (the displacements differed by a mean of less than 0.25 pixels). The reason for this lies in the fact that each stack iteration potentially modifies all the sections and that therefore there is little point in

iterating the force iteration to convergence as the adjacent sections will change. In fact by performing only a few iterations, an intermediate result is generated which may be closer to the final result.

Considering experiments 6.2, 6.5 and 6.7, we see that increasing **FEstop** also causes the result to deteriorate and increases the rate of convergence. The same is true for **warpstop** and **stopcondition**, as can be seen from the results of the group of experiments 6.5, 6.7 to 6.9, and the group 6.5, 6.6, 6.8, and 6.10 respectively.

These results show that although the quality of the reconstruction deteriorates as the convergence parameters are relaxed, the deterioration is gradual. Examination of figure 10.12 which contains the same section in the x-z planes for the results of experiments 6.1, 6.6, and 6.9, shows that there is little visual difference in the quality of the reconstructions. We thus conclude that small changes in the convergence parameters has a gradual effect on the result and that they should be selected to give the required smoothness in the reconstruction.

experiment 6.1

experiment 6.6

experiment 6.9

Figure 10.12: The results of experiments 6.1, 6.6, and 6.9.

10.3.6 Experiment 7: Other Image Sequences

The results of warping data from other series of sections are given in figures 10.13 to 10.15 which show sections cut perpendicularly to the cutting direction, and table 10.8 which show the means and standard deviations of the correspondence alignment measure for the unregistered, registered, and warped stacks. The sets of sections were cut using the same coordinates for each pair.

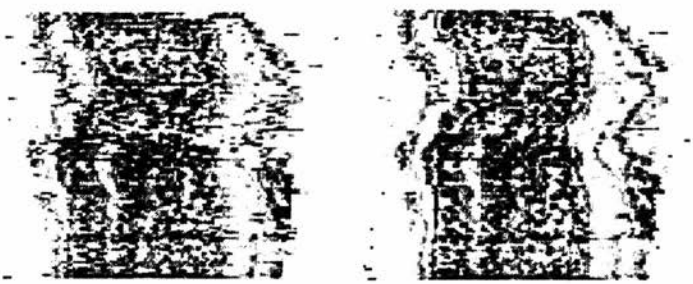


Figure 10.13: The results of warping the 7 day series. The left image shows a section through the original stack; the right image shows the corresponding section (coordinate-wise) through the warped stack. Note that these images were not registered prior to warping.

Figure 10.13 shows the result of warping the 7 day mouse embryo. The left hand image is a section through the original stack; the other image is the corresponding section coordinate-wise through the warped stack. We see that the warping algorithm has smoothed the external edges and the membrane, and has aligned some of the internal structures.

The results of warping the normal and abnormal mouse embryo heart series are given in figures 10.14 and 10.15. Here the top images show a section through the registered stacks, and the bottom images show the corresponding sections through the warped stacks. In both sets of results the main differences are visible in the right hand side of the images where the structure is less complicated. It is hard to see whether there has been any improvement in the alignment of the more complicated structure within the

Figure 10.14: The results of warping the normal heart series. The top image shows a section through the registered stack; the bottom image shows the corresponding section (coordinate-wise) through the warped stack.

Figure 10.15: The results of warping the abnormal heart series. The top image shows a section through the registered stack; the bottom image shows the corresponding section (coordinate-wise) through the warped stack.

	unregistered		registered		warped	
series	mean	std	mean	std	mean	std
7 day	9.671	5.877	—	—	4.636	1.749
9 day	38.545	17.674	11.120	4.780	5.190	0.673
normal heart	29.307	9.530	22.064	9.906	13.077	5.674
abnormal heart	27.282	8.905	16.799	7.756	10.820	4.355

Table 10.8: Table showing the mean and standard deviation of the CAM (pixels^2) values for each series of sections. Note that the 7 day series was not registered before warping.

heart[2]. The poorer results in the left hand side of both heart series is probably due to the complexity of this structure. These reconstructions are also seriously affected by a number of "rogue" sections which have tissue folds or staining artefacts which have not been removed.

The matching algorithm can cope with complicated images, but may be slow to converge because of competition between possible correspondences. Bringing other parts of the sections into alignment should resolve this problem, but this does not seem to have happened for these two series. The reason for this is that the distance penalty parameter, a, discussed in chapter 7 remains constant throughout the warping process. Allowing it to decrease during the calculation would cause more distant possible corresponding lines to be discarded in favour of closer ones. Note that the structures at the sides of the images are not aligned because the interesting points were restricted to the largest circle lying within each image.

If we compare the mean values for the correspondence alignment measure given in table 10.8, we see that the most dramatic improvement of warping over registration has occurred for the 9 and 7 day mouse embryos, with the values for the other sets of data improving less dramatically over registration. This is at least partly because these series were carefully sectioned for reconstruction; the hearts were sectioned for other purposes and were not so carefully prepared.

From these results we conclude that warping produces the results we require and adequately corrects the mismatches remaining after registration.

[2] The ventricles of the heart have a complex internal "spongy" structure known as *traberculations*.

10.3.7 Experiment 8: Approximation by Global Transformations

	7 day		9 day		normal heart		abnormal heart	
transformation	mean	std	mean	std	mean	std	mean	std
rigid	1.206	0.827	0.885	0.698	1.244	0.910	1.031	0.879
affine	0.445	0.337	0.552	0.444	0.758	0.546	0.588	0.455
polynomial	0.332	0.246	0.408	0.335	0.596	0.453	0.455	0.352

Table 10.9: Table showing the mean differences in displacement (in pixels) for each series as the warp transformations are approximated by a rigid, affine and 2nd order polynomial transformations.

The results of approximating the warp transformations, calculated for each of the four series of sections, by rigid, affine and second order polynomial transformations are given in figure 10.16 and table 10.9. These results just show that for all 4 series more parameters improves the fit. However, the mean difference in displacement between the second order polynomial and warp transformations represents significant error. It is possible that a global affine or second order polynomial transformation is sufficient for some sections, but it is not adequate for all sections and therefore local non-linear warp transformations are essential. Furthermore, it cannot be known in advance what the required warp transformation is for each section. To illustrate the need for local warp transformations, an example warp transformation obtained, by the warping algorithm, for each of the four section series is shown in figure 10.17. None of the four displacement maps shown can be replicated using a global transformation. (Note that the displacements are given over the domain of the images to which the warping algorithm was applied, and therefore for registered sections, the displacement maps seem to be at an angle.)

10.4 Summary and Conclusion

The experiments discussed above have shown that warping produces significant improvements over registration and that the resulting reconstruction is smooth. The most important effect of warping is the smoothing which allows proper resolution of structures in arbitrary planes and we conclude that the objectives with respect to the quality of the reconstruction have been achieved. However since the warping process

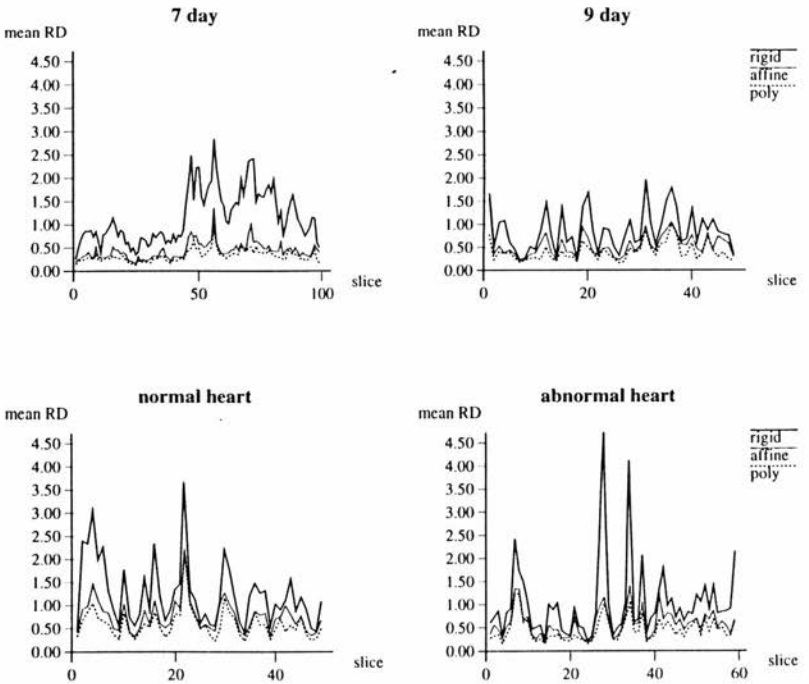
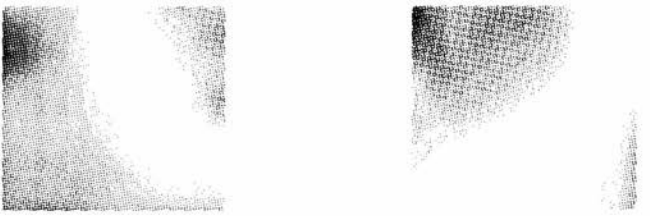


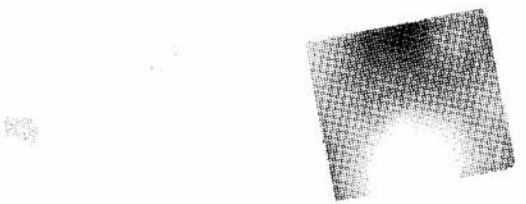
Figure 10.16: Graphs showing the results of approximating the warps by rigid, affine and 2nd order polynomial (poly) transformations. The mean RD is shown in pixels.



7 day mouse embryo

9 day mouse embryo

normal heart



abnormal heart

Figure 10.17: Displacement maps for a section for each of the 4 section series. The x and y displacements are shown on the left and right respectively. Blue shows a negative displacement and red shows a positive displacement. The maximum displacement shown here is 9 pixels. The mean RD is shown in pixels.

does not correct the overall registration of the stack, the best results are achieved with pre-registered sections. We have also established that the warp transformations cannot always be approximated by global transformations within one pixel, and that therefore local warping is necessary.

It was found that fine tuning of the parameters is not necessary in order to produce a satisfactory result. However care is needed when determining **pointdist**, **springk**, and **areasize**, both to ensure convergence of the finite element iteration and to prevent unnecessary computation. **Pointdist** should be chosen sufficiently small to ensure a good result, but if computation time is an issue, the number of interesting points should be restricted. **Springk** should be chosen carefully because if it is too small, the sections will not be sufficiently warped to produce a smooth reconstruction, but if it is too large an inappropriate warp will occur, and there may be convergence problems. To avoid unnecessary computation, **areasize** should be chosen to be the minimum which will enable correct correspondences to be found. We note that **areasize** can be smaller than the maximum expected displacement.

The warping process is slow and requires considerable amounts of memory. The 7 day (100 sections; 1.2 Mb for the stack) and 9 day series (50 sections; 5.5 Mb for the stack) can be warped in 8 to 10 hours on a SPARC 10/41; the two heart series (50 and 60 sections; 6.1 Mb and 8.5 Mb for the normal and abnormal heart series respectively) take 50 to 60 hours. See appendix D for a more detailed analysis of the costs of the warping process.

Chapter 11

Discussion and Conclusion

The primary objective of the work described in this thesis was to develop a fully automatic technique for 3D reconstruction from serial section data. The method had to cope with loss of alignment and sectioning distortions without the use of fiducials, and was required to produce a representative reconstruction of the full 3D voxel image for use in a 3D atlas and gene expression database. We have been completely successful at achieving this objective, and in fact the first complete reconstructions, one from wax embedded material (a 9 day mouse embryo) and one from plastic sections (a 7.75 day embryo) are already in use (see figures 1.1 and 11.1). In reaching this goal novel use has been made of the finite element method and new techniques have been developed in image processing following analysis and experiment for this application.

The main requirements were that the method should be fully automatic and that a full 3D voxel image should result. These requirements are not met by existing published methods and therefore new methods had to be developed. The underlying idea presented in this thesis is to attach springs to correspondences established between sections, where each section is modelled as a thin elastic plate, constrained so that points can move only within the plane of the plate, and then to let go and wait until the system reaches equilibrium.

The finite element method was chosen to model the sections and to calculate the equilibrium position of each section, because it allows arbitrary material properties to be defined in different regions of the images, and the possibility of adding arbitrary constraints provided by the user. The problems of establishing correspondences and of

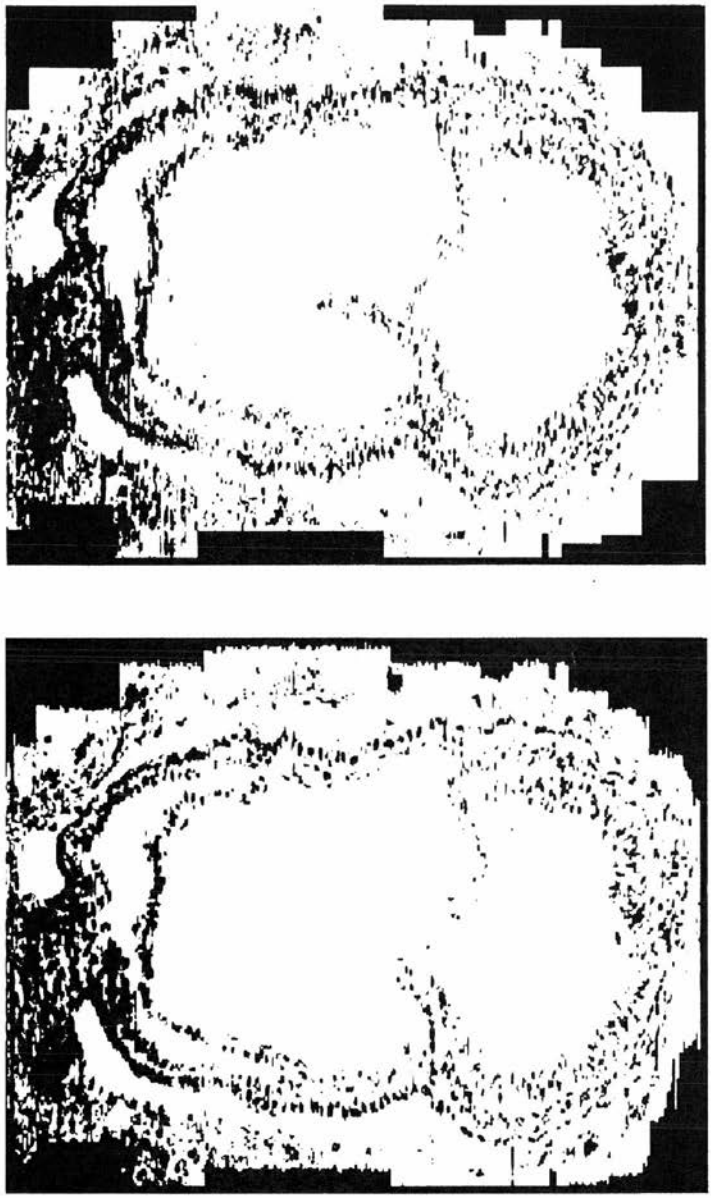


Figure 11.1: Images of a (manually registered) 7.75 day mouse embryo before (top) and after (bottom) warping.

calculating the spring constants were analysed, and new techniques were developed to solve these problems and to establish point to line correspondences. Finally we established the properties and behaviour of the algorithm and showed that its performance was robust with respect to variation in the values of its parameters.

The experiments discussed in chapter 10 provide an analysis of the behaviour of the warping algorithm and show the considerations necessary when applying the technique. Certain aspects of the algorithm can be improved and in this chapter we discuss possibilities for future work to improve the algorithm and to use the techniques developed here for other related applications.

In chapter 10 we showed that the main effect of warping is *smoothing*. Other methods for smoothing reconstructions have been presented in the literature, but most of these require that contours are (manually) extracted [19, 35, 24], and involve *removing data* to smooth the surfaces. The method presented here is fully automatic and does not remove any data. Furthermore, as required by the intended application, smoothing is of the entire greylevel voxel image to enable arbitrary resectioning.

It was also shown in chapter 10 that the warping algorithm is inefficient at correcting registration errors and the final result may depend on the initial positions of the sections. This is not a problem because several registration algorithms described in the literature and evaluated here, have been shown to provide adequate starting points for the warping process.

In fact, it is straightforward to convert the warping algorithm into a registration algorithm to provide a basis for subsequent warping. This can be achieved simply by matching each section to the previous section only (and not the subsequent) and calculating the registration transformation either by using very stiff material properties relative to the spring constants, or by using the least squares method to calculate a rigid transformation from the positions of the correspondences. The former technique would be more efficient as it would enable points to slide along their corresponding lines, reducing the number of force iterations required to bring the sections into alignment. The rigid transformation could be obtained from the nodal displacements.

This approach is similar to that proposed by Toga and Banerjee [99]. Theoretically this

sort of approach based on local matching should give better results than registration methods based on global matching as it should solve the problems of these methods described in chapter 4, but Toga and Banerjee report no improvements in the quality of the reconstruction over global matching methods. The registration algorithm described above may also provide no, or little, improvement in the quality of the reconstruction over global matching methods, but this remains to be tested. If it does not provide a significantly better registration of the sections, it should be discarded in favour of faster algorithms. Note that in common with the method described by Toga and Banerjee, the registration method described above requires that the sections are first roughly registered.

We have found that there are problems of warping complicated structures where there is a lot of competition between edges. The main parameter which governs which edge is chosen is the distance penalty parameter a in equation 8.1 in chapter 8, which determines the weight given to each value in the match map as a function of distance. Clearly this value should depend on the search area, which in turn should be determined by how closely the sections are aligned. We recall from chapter 7 that the force calculation does not give an ideal result in the case where there is competition between corresponding lines, and it was hoped that this problem would be resolved as other parts of the images are brought into alignment. Clearly, if a region of the images contains a lot of detail with a lot of edges this may be slow unless the search area and distance penalty are respectively decreased and increased so that fewer edges compete. This can be achieved by warping in stages. The first stage would be characterised by a large search area, a relatively weak distance penalty, and relatively stiff material properties so that the main features would be brought into alignment. After the first stage had reached equilibrium, subsequent stages would gradually decrease the search area, increase the distance penalty, and use more pliable material properties so that the finer details would also be brought into alignment.

Other methods of resolving ambiguity would include the use of other types of features as well as edges. For example, a corner detector could be used to locate points of high curvature, or clusters of lines could be identified and matched in a way similar to that proposed by Lowe [63]. In fact ideally there should be a library of feature extraction and

matching routines. This will ensure that different types of features can be extracted and matched from the same images, but it will also enable different types of images to be matched. For example, sections where the main structures are individual cells of similar shapes and sizes will require different matching techniques to sections where the structures are of many different sizes and shapes. Note that these extensions would also introduce other types of constraints on the springs. For example, we may require point-to-point as well as point-to-line correspondences.

In the current implementation of the warping algorithm, convergence problems are caused by errors introduced when the images are warped. The number of pixels used to calculate the mean and standard deviation on either side of an hypothesised line is relatively small and changing the value of just a single pixel in the neighbourhood could significantly change the values, which in turn may cause different match values for the points when the section is warped. Using bilinear interpolation rather than nearest neighbour values to calculate the warped images may help, but a better solution would be to improve the feature measurement extraction algorithm. This could be achieved by fitting curves to the means and standard deviations calculated as the mask is rotated, which would enable better approximations of both the line direction and the strength of the edge to be calculated, as all the information would be used rather than just the maximum value of the F test, and the estimate would not suffer as badly from the discretisation.

Another improvement to the warping algorithm would be to improve the interesting point selection procedure so that fewer points in areas of texture are chosen. In the current implementation, the number of points in areas of texture are limited only by the maximum number of points set for each section. If a procedure could be found so that only (or mostly) edge points, including internal edge points, are chosen, the number of interesting points would be better tailored to each individual image. This should also decrease the computation time by reducing the number of correspondences to be established.

In the calculations reported here the material properties were uniform but the finite element method easily extends to elements with differing material properties. However if the background elements are too pliable, alignment of artefacts or mismatched struc

tures could lead to unnecessarily large amounts of warping. Nevertheless experiments can be performed to see if modelling different parts of the images differently based on the grey levels improves the results of warping.

The warping algorithm described in this thesis was designed for small local deformations because it was assumed that the embedding medium would prevent large local deformations caused by, for example, limb articulation. The problems with dealing with large deformations concern problems with the finite element method as it has been implemented. In particular, if a limb has to be bent to bring it into alignment, there will be problems in compressing the elements in the background sufficiently. Other problems may be caused by the fact that the calculated stress and strain become more inaccurate as the mesh is warped as this will have implications on the behaviour of the mesh. However these problems can all be solved: convergence problems can be solved by incrementally increasing the forces and by checking for convergence; the others can be solved by retriangulating the mesh as it is warped, and by removing the background before triangulation, or using non-uniform elasticity.

Although these modifications may not be required for serial section reconstruction, they will be required if the techniques described in this thesis are to be extended to related problems such as warping gene expression data from different sources onto anatomical data as required in the Mouse Atlas Database project [6, 83]. The data will include the expression of various genes during development and it is essential that the different embryos can be matched onto each other[1]. Although this problem is essentially a 3D problem, it should be possible to warp the embryos onto each other section by section. In addition to the problems of coping with large deformations described above, there is the problem of exact matching. For this application it is necessary to warp images so that they become identical to another image and this requires that the elasticity of the material should tend to zero.

The work presented in this thesis provides a good foundation for 2D warping and in principal the techniques can be extended to full 3D warping. This would introduce new

[1] Sections can be stained for both anatomy and gene expression, and the different stains can be viewed separately by the use of filters or other microscopical techniques. Therefore gene expression data can be warped by first matching and warping the anatomical images, and new matching techniques do not have to be developed.

and interesting problems. For example, one application requiring 3D matching and warping is interpolation between different stages of embryonic development. Interpolation between developmental stages is required so that a continuous picture of embryonic development can be obtained, and to cope with the fact that, for example, all the 7 day embryos will not all be at exactly the same stage. This will ultimately include modelling the development of the embryo itself by modelling the specific properties and behaviour of each tissue (see for example Clausi and Brodland [20] who investigate the forces required to ensure closure of the neural tube.)

In the light of the above discussion we summarise possible future work as follows:

- Convert the warping algorithm to a registration algorithm as described above to see if this approach is better than published registration methods.
- Gradually decrease the search area, tighten the distance penalty, and make the sections less stiff as the sections are brought into alignment. This should help to resolve instances where there is much competition between possible corresponding edges.
- Introduce and incorporate other feature extraction and matching routines and improve the interesting point finding and finite element mesh calculation algorithms.
- Enable the warping algorithm to cope with sections containing large amounts of local deformation. This includes ensuring that the finite element iteration converges.
- Experiment with different material properties in different parts of the images
- Extend the techniques developed in this thesis to the problem of 3D warping
- Explore the use of 3D warping to model time development between different embryonic stages and morphogenesis.

The latter extensions take the image processing methods into the realm of theoretical biology and would enable testing of computational models of gene behaviour and interactions, and embryo development, which is the long term goal of the Mouse Atlas project.

11.1 Conclusion

We have shown that the approach taken in this thesis to solve the problem of automatic three dimensional reconstruction from serial histological sections produces satisfactory results. The best results are obtained for pre-registered sections since warping does not correct the overall registration of the stack. However registration algorithms from the literature have been presented and analysed, and several of these are suitable for registering the sections prior to warping.

In order to solve the warping problem, the sections were modelled as thin elastic plates. Springs were attached from "interesting points" on each section to corresponding lines in adjacent sections and the system was allowed to reach equilibrium. To do this, a numerical method for modeling the sections and calculating their final configurations as the forces were applied had to be found. The finite element method was chosen for this task as it enables different parts of the sections to be modelled differently and allows arbitrary constraints such as fixed points or lines to be added or deleted simply. The problems of calculating the correspondences and determining the forces required several new techniques to be developed. Since other methods of feature extraction and matching were found to be unsuitable, a new image filter which calculates the mean and standard deviation on either side of an hypothesised line was developed and a new matching algorithm was devised based on these feature attributes. The conflicting requirements of correspondence and warping were overcome in the development of a method for finding corresponding lines which assigns weights which do not depend on distance. Finally a measure for measuring the alignment of a stack of images was proposed and tested.

Although the warping algorithm was found to be successful, and can be applied to other series of sections without fine tuning the parameters, ideas for improvements in many aspects of the algorithm have been presented and discussed. The most interesting goal is to extend the work to the related problems of mapping gene expression data onto the digital atlas for the gene expression database, and ultimately to modelling the development of the embryo itself.

Bibliography

[1] A. Andreason, A. M. Drewes, J. E. Assentoft, and N. E. Larsen. Computer-assisted alignment of standard serial sections without use of artificial landmarks. A practical approach to the utilization of incomplete information in 3-D reconstruction of the hippocampal region. *Journal of Neuroscience Methods*, 45(3):199 207, 1992.

[2] A. Apicella, J. S. Kippenhan, and J. H. Nagel. Fast multi-modality image matching. *SPIE vol 1092 Medical Imaging III: Image Processing*, pages 252 263, 1989.

[3] N. Ayache and O. D. Faugeras. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Pami-8(1):44–54, January 1986.

[4] N. Baba and K. Kanaya. Three-dimensional reconstruction from serial section images by computer graphics. In Scanning Microscopy, Supplement 2, pages 303-314, 1988.

[5] R. Bajcsy and S. Kovačič. Multiresolution elastic matching. *Computer Vision, Graphics and Image Processing*, 46:1–21, 1989.

[6] R. Baldock, J. Bard, M. Kaufman, and D. Davidson. A real mouse for your computer. *BioEssays*, 14(7), July 1992.

[7] K. Bartels, A. Bovik, S. J. Aggarwal, and K. R. Diller. The analysis of biological shape changes from multidimensional dynamic images. *Computerized Medical Imaging and Graphics*, 17(2):89–99, 1993.

[8] K.-J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice Hall, 1982.

[9] C.-H. Berthold, O. Corneliuson, and M. Rydmark. Changes in shape and size of cat spinal root myelinated nerve fibers during fixation and vestopal W embedding for electron microscopy. *Journal of Ultrastructure Research*, 80:23–41, 1982.

[10] C.-H. Berthold, M. Rydmark, and O. Corneliuson. Estimation of sectioning compression and thickness of ultrathin sections through vestopal-W-embedded cat spinal roots. *Journal of Ultrastructure Research*, 80:42–52, 1982.

[11] T. W. Blackstad and J. G. Bjaalie. Computer programs for neuroanatomy: Three dimensional reconstruction and analysis of populations of cortical neurons and other bodies with a laminar distribution. *Computers in Biology and Medicine*, 18(5):321–340, 1988.

[12] R. C. Bolles and R. A. Cain. Recognising and locating partially visible objects: The local-feature-focus method. *International Journal of Robotics Research*, 1:57–82, 1982.

[13] F. L. Bookstein. Principal warps: Thin plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[14] F. L. Bookstein. Thin plate splines and the atlas problem for biomedical images. *Lecture Notes in Computer Science*, 511:326–342, 1991.

[15] R. A. Brooks. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, 17:285–348, 1981.

[16] R. A. Brooks. Model-based three-dimensional interpretations of two dimensional images. In M. A. Fishler and O. Firschein, editors, *Readings in Computer Vision*, pages 360–370. Morgan Kaufman, 1987.

[17] D. J. Burr. A dynamic model for image registration. *Computer Graphics and Image Processing*, 15:102–112, 1981.

[18] D.J. Burr. Elastic matching of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(6):708–713, 1981.

[19] I. Carlbom, D. Terzopoulos, and K. M. Harris. Computer-assisted registration, segmentation, and 3D reconstruction from images of neuronal tissue sections. *IEEE Transactions on Medical Imaging*, 13(2):351–362, 1994.

[20] D. A. Clausi and G. W. Brodland. Mechanical evaluation of theories of neurulation using computer simulations. *Development*, 118:1013–1023, 1993.

[21] L. D. Cohen and I. Cohen. A finite element method applied to new active contour models and 3D reconstruction from cross sections. Technical Report 1245, INRIA, 1990.

[22] T. F. Cootes and C. J. Taylor. Active shape model search using local grey level models: A quantitative evaluation. In *Proceedingings of the 4th British Machine Vision Conference*, pages 639–648, September 1993.

[23] M. H. Deverell, N. Bailey, and W. F. Whimster. Tissue distortion in three-dimensional reconstruction of wax or plastic embedded microscopic structures. *Pathology Research and Practise*, 185:598–601, 1989.

[24] M. H. Deverell, J. R. Salisbury, M. J. Cookson, J. G. Holman, E. Dykes, and W. F. Whimster. Three-dimensional reconstruction: Methods of improving image registration and interpretation. *Analytical Cellular Pathology*, 5:253–262, 1993.

[25] N. Dodd. Multispectral texture synthesis using fractal concepts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):703–707, September 1987.

[26] R. Dürr, E. Peterhans, and R. von der Heydt. Correction of distorted images pairs with elastic models. *European Journal of Cell Biology*, 48:85–88, 1989. Supplement 25.

[27] B. C. Erricker. *Advanced General Statistics.* Hodder and Stoughton, 1971

[28] J.-Q. Fang and T. S. Huang. Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(5):545–554, 1984.

[29] R. B. Fisher. *From Surfaces to Objects. Computer Vision and Three Dimensional Scene Analysis.* John Wiley & Sons, 1989.

[30] A. R. Forsyth. *Calculus of Variations.* Dover Publications, 1960.

[31] P. N. Gaunt and W. A. Gaunt. *Three Dimensional Reconstruction in Biology.* Pitman Medical, 1978.

[32] C. Goad. Fast 3D model-based vision. In A. P. Pentland, editor, *From Pixels to Predicates*, chapter 16, pages 371–391. Ablex, 1986.

[33] R. C. Gonzalez and P. Wintz. *Digital Image Processing.* Addison Wesley, second edition, 1987.

[34] A. Goshtasby, D. A. Turner, and L. V. Ackerman. Matching of tomographic slices for interpolation. *IEEE Transactions on Medical Imaging*, 11(4):507–516, December 1992.

[35] H. M. Gérard, J. C. André, and J. L. Mallet. Computer-Aided 3-D reconstructions from routine histologic sections. *Microscopy and Analysis*, pages 31–33, January 1993.

[36] W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, 1990.

[37] T. D. Haig, Y. Attikiouzel, and M. Alder. Border marriage: Matching of contours of serial sections. *IEE Proceedings Communications Speech and Vision*, 138(5):371–376, 1991.

[38] B. Heppelmann, K. Meßlinger, and R. F. Schmidt. Serial sectioning, electron microscopy, and three dimensional reconstruction of fine nerve fibres and other extended objects. *Journal of Microscopy*, 156:163–172, November 1989.

[39] M. Herbin, A. Venot, J.Y. Devaux, E. Walter, J.F. Lebruchec, L. Dubertret, and J.C. Roucayrol. Automated registration of dissimilar images: Application to medical imagery. *Computer Vision, Graphics and Image Processing*, 47:77–88, 1989.

[40] L. S. Hibbard, T. L. Arnicar-Sulze, B. J. Dovey-Hartman, and R. B. Page. Computed alignment of dissimilar images for three dimensional reconstructions. *Journal of Neuroscience Methods*, 41:133–152, 1992.

[41] L. S. Hibbard, B. J. Dovey-Hartman, and R. B. Page. Three-dimensional reconstruction of median eminence microvascular modules. *Computers in Biology and Medicine*, 16(6):411–421, 1986.

[42] L. S. Hibbard, R. A. Grothe, and T. L. Arnicar-Sulze. Computed alignment of serial sections for 3-D reconstructions. In *SPIE*, volume 1905, pages 946–955, 1993.

[43] L. S. Hibbard and R. A. Hawkins. Objective image alignment for three-dimensional reconstruction of digital autoradiograms. *Journal of Neuroscience Methods*, 26:55–74, 1988.

[44] A. Hill, T. F. Cootes, C. J. Taylor, and K. Lindley. Medical image interpretation: A generic approach using deformable templates. *Medical Informatics*, 19(1):47–59, 1994.

[45] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[46] D. P. Huijsmans, W. H. Lamers, J. A. Los, and J. Strackee. Toward computerized morphometric facilities: A review of 58 software packages for computer-aided three-dimensional reconstruction, quantification, and picture generation from parallel serial sections. *The Anatomical Record*, 216:449–470, 1986.

[47] D.P. Huijsmans, W. H. Lamers, J. A. Los, J. Smith, and J. Strackee. Computer-aided three-dimensional reconstruction from serial sections: A software package for reconstruction and selective image generation for complex topologies. In K. Ko and H. A. Tucker, editors, *Eurographics '84*, pages 3–13. Elsevier Science Publishers, 1984.

[48] D. P. Huttenlocher and S. Ullman. Recognising solid objects by alignment. In *Proceedings DARPA Image Understanding Workshop*, pages 1114–1124, 1988.

[49] T. Jansson. Computerized alignment and deformation correction of microscopic serial section images of nervous tissue. Technical Report 257, School of Electrical and Computer Engineering, Chalmers University of Technology, Göteborg, Sweden, 1994.

[50] G. A. Johnson, H. Benveniste, R. D. Black, L. W. Hedlund, R. R. Maronpot, and B. R. Smith. Histology by magnetic resonance microscopy. *Magnetic Resonance Quarterly*, 9(1):1–30, 1993.

[51] A. S. Jones, B. K. Milthorpe, and C. R. Howlett. Measurement of microtomy induced section distortion and its correction for 3-dimensional histological reconstructions. *Cytometry*, 15:95–105, 1994.

[52] M. M. Jordan and W. J. Perkins. An expert system approach to the identification and categorisation of features of biological images. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition and Artificial Intelligence*, pages 461–470. Elsevier Science Publishers, 1988.

[53] R. D. Kapadia, W. B. High, H. A. Soulleveld, D. Bertolini, and S. K. Sarkar. Magnetic resonance microscopy in rat skeletal research. *Magnetic Resonance in Medicine*, 30(2):247–250, 1993.

[54] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.

[55] M. H. Kaufman. *The Atlas of Mouse Development.* London: Academic Press., 1992.

[56] J. M. Keller, S. Chen, and R. M. Crownover. Texture description and segmentation through fractal geometry. *Computer Vision, Graphics and Image Processing*, 45:150–166, 1989.

[57] A. Kriete. Methodological basis of three dimensional imaging and visualization in biomedical sciences a review. In *MedTech '89: Medical Imaging*, pages 2 13, November 1989. SPIE Volume 1357.

[58] A. Kriete and G. Magdowski. Computerised three dimensional reconstructions of serial sections in electron microscopy. *Ultramicroscopy*, 32:48–54, 1990.

[59] A. C. Laan, W. H. Lamers, D. P. Huijsmans, A. te Kortschot, J. Smith, J. Strackee, and J. A. Los. Deformation-corrected computer-aided three-dimensional reconstruction of immunohistochemically stained organs: Application to the rat heart during early organogenesis. *The Anatomical Record*, 224:443–457, 1989.

[60] W. H. Lamers, A. C. Laan, D. P. Huijsmans, J. Smith, J. A. Los, and J. Strackee. Deformation corrected computer aided 3 D reconstruction of immunohistochem ically stained sections of embryonic organs. *European Journal of Cell Biology*, 48:103–106, 1989. Supplement 25.

[61] D. T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer and Information Sciences*, 9(3):217 242, 1980.

[62] J. Lockhausen, U. Kristen, W. Menhardt, and W. J. Dallas. Three-dimensional reconstruction of a plant dictyosome from series of ultrathin sections using com puter image processing. *Jounal of Microscopy*, 158(2):197–205, May 1990.

[63] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[64] A. Macovski. *Medical Imaging Systems.* Information and Systems Science. Prentice-Hall, 1983.

[65] M. Merickel. 3D reconstruction: The registration problem. *Computer Vision, Graphics, and Image Processing*, 42:206–219, 1988.

[66] E. P. Meyer and V. J. Domanico. Three-dimensional reconstruction: a tissue embedding method for alignment of serial sections. *Journal of Neuroscience Methods*, 26:129 132, 1988.

[67] R. S. Mitra and N. N. Murthy. Elastic, maximal matching. *Pattern Recognition*, 24(8):747–753, 1991.

[68] M. Moshfeghi. Elastic matching of multimodality medical images. *CVGIP: Graphical Models and Image Processing*, 53(3):271–282, May 1991.

[69] V. A. Moss, D McEwan Jenkinson, and H. Y. Elder. Automated image segmentation and serial section reconstruction in microscopy. *Jounal of Microscopy*, 158(2):187–196, May 1990.

[70] P. H. Mowforth and J. Zhengping. Model based tissue differentiation in MR brain images. In *Proceedings of the Fifth ALVEY Vision Conference*, pages 67–71, 1989.

[71] R. W. Ogden. *Non-Linear Elastic Deformations.* Ellis Horwood, 1984.

[72] J.-C. Olivo, J.-C. Izpisúa-Belmonte, C. Tickle, C. Boulin, and D. Duboule. Reconstruction from serial sections: a tool for developmental biology. application to hox genes expression in chicken wing buds. *Bioimaging*, 1:151–158, 1993.

[73] D. R. J. Owen and E. Hinton. *A Simple Guide to Finite Elements.* Pineridge Press, 1980.

[74] D. A. Perednia and R. G. White. Automatic registration of multiple skin lesions by use of point pattern matching. *Computerized Medical Imaging and Graphics*, 16(3):205–216, 1992.

[75] W. J. Perkins, M. M. Jordan, and A. M. Shepherd. Human-computer interaction and expert systems for three-dimensional studies of biomedical images. *Medical Informatics*, 14(2):135–147, 1989.

[76] J. Piper and D. Rutovitz. Data structures for image processing in a C language and unix environment. *Pattern Recognition Letters*, 3:119–129, 1985.

[77] I. Poole, A. Carothers, and D. Charleston. Object relocation in automated image cytometry. *Bioimaging*, 2:62–68, 1994.

[78] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, 1985.

[79] J. S. Prothero and J. W. Prothero. Three-dimensional reconstruction from serial sections IV. The reassembly problem. *Computers and Biomedical Research*, 19:361–373, 1986.

[80] J. S. Prothero and J. W. Prothero. A software package in C for interactive 3-D reconstruction and display of anatomical objects from serial section data. In *NCGA '89 Conference Proceedings. 10th Annual Conference and Exposition Dedicated to Computer Graphics*, pages 187–192, 1989.

[81] D. W. A. Rees. *Mechanics of Solids and Structures.* McGraw-Hill, 1990.

[82] J. A. Richards, F. W. Sears, M. R. Wehr, and M. W. Zemansky. *Modern University Physics.* Addison Wesley, 1960.

[83] M. Ringwald, R. Baldock, J. Bard, M. Kaufman, J. T. Eppig, J. E. Richardson, J. H. Nadeau, and D. Davidson. A database for mouse development. *Science*, 265:2033–2034, September 1994.

[84] N. W. Runham, D. A. Davies, and D. Roberts. Computer-aided three dimensional reconstruction from serial sections. *Microscopy and Analysis*, pages 15–18, March 1990.

[85] M. Rydmark, T. Jansson, C.-H. Berthold, and T. Gustavsson. Computer-assisted realignment of light micrograph images from consecutive section series of cat cerebral cortex. *Journal of Microscopy*, 165(1):29–47, January 1992.

[86] E. M. Santori and A. W. Toga. Superpositioning of 3-dimensional neuroanatomic data sets. *Journal of Neuroscience Methods*, 50:187–196, 1993.

[87] C. Schnörr. Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class. *International Journal of Computer Vision*, 6(1):25–38, 1991.

[88] S. Sclaroff and A. Pentland. Modal matching for correspondence and recognition. Technical Report 201, M.I.T. Media Laboratory Perceptual Computing Section, May 1993.

[89] G. L. Scott and H. C. Longuet-Higgings. An algorithm for associating the features of two images. *Proceedings of the Royal Society of London*, B244:21–26, 1991.

[90] L. S. Shapiro and J. M. Brady. A modal approach to feature-based correspondence. In P. Mowforth, editor, *British Machine Vision Conference*, pages 78–85, 1991.

[91] D. Skea, I. Barrodale, R. Kuwahara, and R. Poeckert. A control point matching algorithm. *Pattern Recognition*, 26(2):269–276, 1993.

[92] F. L. Stasa. *Applied Finite Element Analysis for Engineers*. CBS International Editions, 1985.

[93] M. J. Stephens, R. J. Blisset, D. Charnley, E. P. Sparks, and J. M. Pike. Outdoor vehicle navigation using passive 3D vision. In *IEEE Proceedings: Computer Vision and Pattern Recognition*, pages 556–562, 1989.

[94] G. Sullivan, P. Karaolani, and C. Attwood. Model-based processing of radiological images. Alvey MMI-134 University of Reading Final Report., 1991.

[95] M. Sun. Alignment of serial sections for 3D visualization. Master's thesis, University of Pittsburgh, School of Engineering, 1986.

[96] M. Sun and C. C. Li. Registration of serial sectional images for 3 D reconstruction. In *Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics*, pages 992–996, 1986.

[97] A. W. Toga and T. L. Arnicar. Image analysis of brain physiology. *IEEE Computer Graphics and Applications*, 5:20–25, 1985.

[98] A. W. Toga and T. L. Arnicar-Sulze. Digital image reconstruction for the study of brain structure and function. *Journal of Neuroscience Methods*, 20:7–21, 1987

[99] A. W. Toga and P. K. Banerjee. Registration revisited. *Journal of Neuroscience Methods*, 48:1–13, 1993.

[100] F. J. Verbeek. Deformation correction using euclidean contour distance maps. In *Proceedings of the 11th IAPR International Conference on Pattern Recognition Vol III Image, Speech and Signal Analysis*, pages 347–350, 1992.

[101] M. Vuillemin, T. Pixieder, Y. M. Wong, and R. P. Thompson. A two-step alignment method for 3D computer aided reconstruction based on fiducial markers and applied to mouse embryonic hearts. *European Journal of Morphology*, 30(3):181–193, 1992.

[102] R. G. White and D. A. Perednia. Automatic derivation of initial match points for paired digital images of skin. *Computerized Medical Imaging and Graphics*, 16(3):217–225, 1992.

[103] J.O.N. Woodward. Change detection in MRI brain scan data. Master's thesis, Department of Artificial Intelligence, University of Edinburgh, 1990.

[104] H. Yaegashi, T. Takahashi, and M. Kawasaki. Microcomputer-aided reconstruction: a system designed for the study of 3 D microstructure in histology and histopathology. *Journal of Microscopy*, 146(1):55–65, 1987.

[105] Y. J. Zhang. Automatic correspondance finding in deformed serial sections. In E. J. Karjalainen, editor, *Scientific Computing and Automation (Europe) 1990*, pages 39–53. Elsevier, 1990.

[106] Y. J. Zhang. 3-D image analysis system and megakaryocyte quantitation. *Cytometry*, 12:308–315, 1991.

[107] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*, volume 1. McGraw-Hill, 4th edition, 1989.

[108] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*, volume 2. McGraw-Hill, 4th edition, 1991.

Appendix A

Glossary

alignment Bringing individual image structures, which may represent fiducials, into line with each other. This may involve non rigid transformations.

contour The boundaries of an anatomical structure in a section which have been found interactively or automatically.

deformation or distortion A section is deformed with respect to the previous section if the difference between the sections can not be corrected using rotation and translation. Deformations can be the result of compression or stretching in one or more directions but tend to be local and random.

feature Any image structure in the section which is useful in registering and warping the section images such as contours, lines, points, or regions. Features may represent fiducial markers or anatomical structures such as, for example, main blood vessels and main nerve fibres.

feature attribute Any measurement that can be taken from a feature, for example length, area, perimeter and compactness.

interesting point One of a set of points chosen from one image to match to another image in order to establish correspondences.

object A structure in an image which is completely surrounded by background, or a collection of such structures. The image of all the tissue contained within a single section can also be referred to as an object.

matchmap The image produced by matching an "interesting" point in one image to a region in another image. The pixel values indicate how well the point matches each point in the other image.

registration A process which corrects for the rotational and translational misalignment in the sections by rigid transformations.

stack A 3D image obtained by assembling a series of 2D sections.

warping A process which corrects for the deformations in the sections.

Appendix B

Glossary of Symbols for Finite Element Analysis

Symbol	Description
$\Delta\mathbf{A}^{(i)}$	vector of increments of the nodal point displacement in iteration i
$\mathbf{a}$	vector of nodal displacements
$\mathbf{B}$	strain-displacement transformation matrix for linear analysis
${}^{t}_{0}\mathbf{B}_{\mathbf{L}}$	linear strain-displacement transformation matrix
${}^{t}_{0}\mathbf{B}_{\mathbf{NL}}$	non-linear strain-displacement transformation matrix
${}_{0}\underline{\mathbf{C}}$	incremental stress-strain material property tensor referred to the configuration at time zero
${}_{0}\mathbf{C}$	incremental stress-strain material property matrix referred to the configuration at time zero
${}_{0}C_{ijkl}$	components of the incremental stress-strain material property tensor referred to the configuration at time zero
$\mathbf{D}$	linear constitutive matrix
$\mathbf{d}^{i}$	vector describing the displacement of the point to which spring i is attached
$\mathcal{E}$	energy functional
E	Young's modulus
${}_{0}e_{ij}$	components of the linear incremental strain tensor referred to the configuration at time zero
$\mathbf{F}_{\mathbf{s}}$	force vector due to springs
$\mathbf{F}_{\mathbf{I}}$	vector of initial forces
$\mathbf{F}_{\mathbf{p}}$	force vector due to stresses in the mesh
$\mathbf{F}_{\mathbf{s}}$	force vector due to springs
$\mathbf{f}^{i}$	vector describing the initial extension of spring i
$\underline{\mathbf{G}}$	Green-Lagrange strain tensor
G_{ij}	components of Green-Lagrange strain tensor
$\mathbf{g}_{i}$	eigen vectors of Green-Lagrange strain tensor
g_{i}	eigen values of Green-Lagrange strain tensor
h	element thickness
J	local ratio of current to reference volume
$\mathbf{K}$	stiffness matrix for the mesh

$\mathbf{K}_s$	stiffness matrix for the springs
$\mathbf{K}^i$	stiffness matrix for spring i
$\mathbf{N}^e$	element shape function matrix
N_i^e	element shape function
p	hydrostatic part of stress in an incompressible elastic material
$\underline{\mathbf{S}}$	2nd Piola-Kirchhoff stress tensor
S_{ij}	components of the 2nd Piola-Kirchhoff stress tensor
$\mathbf{S}$	matrix corresponding to 2nd Piola-Kirchhoff stress tensor
$\hat{\mathbf{S}}$	vector corresponding to 2nd Piola-Kirchhoff stress tensor
s_i	principal components of the 2nd Piola-Kirchhoff stress tensor
$\mathbf{u}$	displacement vector for the mesh
$\mathbf{u}^e$	displacements inside element e
u_i	components of the displacement tensor
W	strain energy function
α_p	material constants
Δ	area of element
δ_{ij}	Kronecker delta
$\boldsymbol{\epsilon}$	engineering strain vector
ϵ_{ij}	components of the engineering strain tensor
${}_0\eta_{ij}$	components of the non-linear incremental strain tensor referred to the configuration at time zero
λ_i	principal stretches
μ_p	material constants
ν	Poisson's ratio
${}^t\rho$	mass density at time t
$\boldsymbol{\sigma}$	engineering stress vector
σ_{ij}	components of the engineering stress tensor

Appendix C

Attaching Springs to Infinite Lines

We derive the stiffness matrix $\mathcal{K}^i$ and the vector describing the initial extension of the spring $\mathbf{f}^i$ of expression 6.2 in the case where spring i is attached to an infinite line.

Let the initial position of the end of spring i attached to the section to be warped be $\mathbf{q}$, the displacement of this point be $\mathbf{d}$, and the equation of the corresponding line be $\mathbf{l} = \mathbf{t} + \lambda\mathbf{a}$, where t is a point on the line and $\mathbf{a} = [a_x, a_y]^T$ is the unit direction vector of the line (see figure C.1). The initial force is that generated by the spring when the

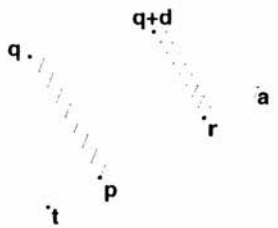


Figure C.1: Allowing points to slide along lines.

end of the spring attached to the line is at $\mathbf{p}$ where $\mathbf{p}$ lies on the corresponding line, and $\overline{\mathbf{pq}}$ is perpendicular to the line. The initial force can be written as

$$\mathbf{F}_\mathbf{q} = k\,\mathbf{f}^i = k(\mathbf{p} - \mathbf{q}),$$

where k is the spring constant and it can be shown that $\mathbf{p}$, the closest point on the line to $\mathbf{q}$, is

$$\mathbf{p} = \mathbf{t} + [(\mathbf{q} - \mathbf{t}) \cdot \mathbf{a}]\,\mathbf{a}$$

Similarly the closest point on the line to $\mathbf{q} + \mathbf{d}$ is

$$\mathbf{r} = \mathbf{p} + [((\mathbf{q} + \mathbf{d}) - \mathbf{p}) \cdot \mathbf{a}]\,\mathbf{a},$$

and the force exerted by the spring when the end attached to the section is at $\mathbf{q} + \mathbf{d}$ is

$$\mathbf{F}_{\mathbf{q}+\mathbf{d}} = k(\mathbf{r} - (\mathbf{q} + \mathbf{d}))$$

which can be written as

$$\mathbf{F}_{q+d} = \mathcal{K}^i(\mathbf{f}^i - \mathbf{d}),$$

where

$$\mathcal{K}^i = -k \begin{pmatrix} a_x^2 - 1 & a_x a_y \\ a_x a_y & a_y^2 - 1 \end{pmatrix}.$$

It was found that attaching the springs to infinite lines caused convergence problems. This was solved by attaching each spring to the nearest point on its corresponding line, and updating this attachment point at each finite element iteration. In this case, $\mathbf{f}^i$ is as above,

$$\mathcal{K}^i = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix},$$

and the updated initial force vector is given by

$$\mathbf{F}_{q+d} = -k \begin{pmatrix} a_x^2 - 1 & a_x a_y \\ a_x a_y & a_y^2 - 1 \end{pmatrix} (\mathbf{f}^i - \mathbf{d}),$$

where $\mathbf{d}$ denotes the current displacement for the point $\mathbf{q}$.

Appendix D

Implementation and Timing

Various implementation details have been given throughout the thesis where they relate to the methods being described. Here we discuss details of the implementation of the warping algorithm which are not central to the ideas and methods described in the thesis. We also show the flow of control in the warping program and in the procedure used to warp each section. This is followed by a brief analysis of the time spent at each stage of the process.

D.1 Implementation

The warping algorithm was implemented in C and with the exception of the Delaunay triangulation routines and some of the image processing routines, all the code was written specifically for this application.

All the image processing was implemented using the woolz image processing library and structures [76]. Instead of storing images as a rectangular array of values, this library represents images as a series of "intervals" (one dimensional arrays of contiguous pixel values), which allows an efficient representation of images of arbitrary domain, and thus reduces the time required to process such images.

The finite element mesh was obtained from the Delaunay triangulation of a set of points. The Delaunay triangulation routines were supplied by Glynn Robinson (Dept of Diagnostic Radiology, Yale University) and Lewis Griffin (Dept of Neurology, UMDS, University of London). The structures used by these algorithms were converted to the structures used in the finite element calculations.

All the finite element analysis routines were implemented using the representation of the stiffness matrix, designed for symmetric banded matrices, recommended in the book by Bathe [8]. The finite element equations were solved using Gaussian elimination as described by Bathe for this representation.

D.2 Structure of the Warping Program

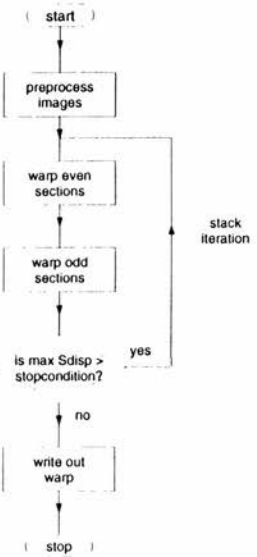


Figure D.1: The top level control of the warping program showing the stack iteration.

As described in chapter 8, there are three levels of iteration in the warping algorithm: stack iteration, force iteration and FE iteration. The Stack iteration is depicted in figure D.1 which shows the process of warping the whole stack by warping alternate sections using the most recent information (iteration scheme 1 in section 8.4). In this figure stopcondition is one of the convergence parameters defined in chapter 10: warping is deemed to have converged when the maximum displacement (Sdisp) for a stack iteration is less than stopcondition (default value: 0.8 pixels).

An overview of the procedure employed to warp a single section is given in figure D.2 where the force and FE iterations are shown. The various variables referred to in this figure are the convergence parameters discussed in chapter 10 and variables to denote various displacements (default values are given in square brackets):

FEstop [0.2] (finite element iteration): Stop the finite element iteration when the maximum displacement calculated during a single iteration (FEdisp) is less than this value.

warpstop [0.5] (force iteration): stop the force iteration when the maximum displacement calculated during a single force iteration (Fdisp) is less than this value.

recalc [5.1] (force iteration): recalculate the image and the feature measurements when the maximum displacement since the last time the image was recalculated (Idisp) exceeds this value.

maxiters [11] (force iteration): Maximum number of iterations for the force iteration.

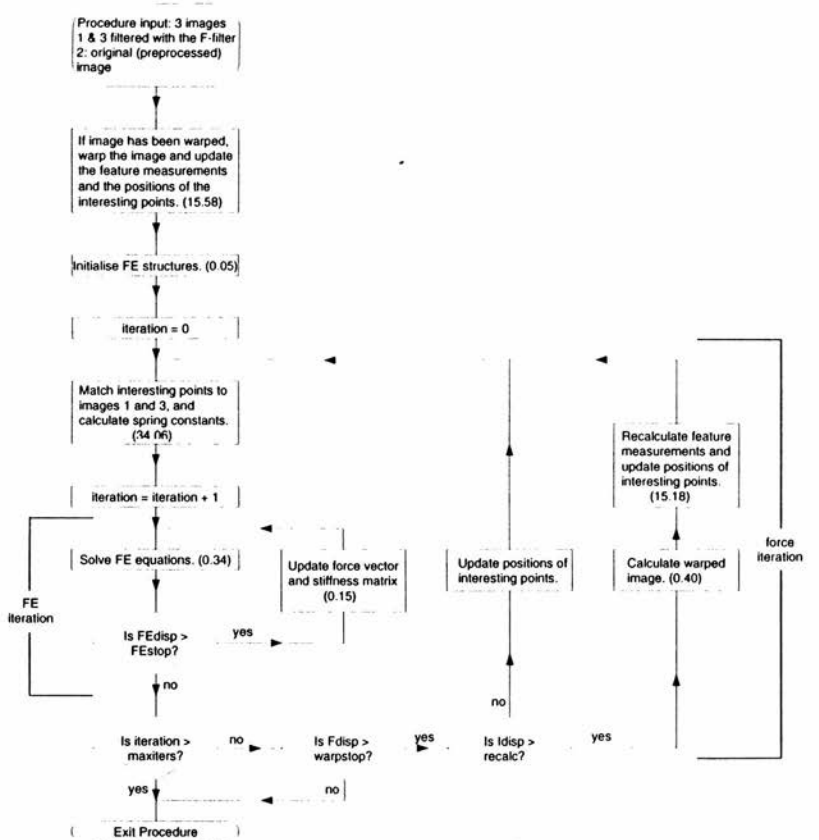


Figure D.2: The process used to warp a single image showing the FE and force iterations. The numbers in brackets give the approximate number of seconds needed to complete that stage.

As discussed in chapter 8, the image does not need to be recalculated during each force iteration since significant changes in the feature measurements associated with the interesting points only occur when the hypothesised line direction changes. This line direction will only change significantly under a significant rotation, and since groups of pixels cannot undergo significant rotations without affecting other parts of the image, a fairly large threshold can be set. A maximum number of force iterations is set to prevent oscillation.

D.3 Timing

At the end of chapter 10 we stated the amount of time required (using a SPARC 10/41) to warp each of the 4 sets of serial sections on which the warping algorithm was tested. This varied from 8 10 hours to 50 60 hours depending on the size and number of the images and the amount of warping required to smooth the resulting reconstruction. Here we analyse the time spent warping a single image.

One of the images from the 9 day series was selected and a test stack was made by sandwiching a (manually) warped copy of this image (warp 1 of chapter 10 was used) between two copies of the original image. The warping program was applied to this stack using the default parameters given in chapter 10 and the time spent in each routine was analysed using Quantify[1]. Note that the number of stack iterations was limited to 1; the embryo was extracted from the images reducing the number of pixels to be analysed from 53 766 to 41 565 for the original image, and from 54 668 to 42 157 for the warped image; and 500 interesting points were found.

Warping was completed after 38 FE iterations and 6 force iterations where the middle image was recalculated 3 times. The time in seconds required to complete each stage of the warping process is shown in figure D.2. No time is given for those stages taking less than 0.01 seconds. From this figure we may conclude that most of the time is spent doing image processing: 15.18 seconds are required to calculated the feature measurements for each pixel position in an image; 31 seconds are required to match 500 interesting points to two images. In contrast, a single finite element iteration requires only 0.19 seconds. In total, approximately 93.4% of the time spent warping a single image is spent doing image processing against 6.6% for finite element analysis.

The reason for the large amount of time spent doing image processing is due to the techniques used: 12.5×5 masks are passed over an image in order to calculate the feature measurements; and the forces are calculated by matching each interesting point to each adjacent image 25 times.

[1] Quantify: Pure Software Inc. 1309 South Mary Avenue, Sunnyvale, CA 94087, USA.